

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
(інститут)

Факультет інформаційних технологій
(факультет)

Кафедра Програмного забезпечення комп'ютерних систем
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня
бакалавра

(назва освітньо-кваліфікаційного рівня)

студента *Скітенко Анастасії Юріївни*
(ПІБ)

академічної групи *122-19-4*
(шифр)

спеціальності *122 Комп'ютерні науки*
(код і назва спеціальності)

освітньої програми
(назва освітньої програми)

на тему: *Розробка веб-орієнтованого застосунку для
астрономічної обсерваторії засобами мов HTML, CSS, JavaScript та PHP*

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	<i>доц. Спірінцев В.В.</i>			
розділів:				
спеціальний	<i>доц. Спірінцев В.В.</i>			
економічний	<i>доц. Вагонова О.Г.</i>			
Рецензент				
Нормоконтролер	<i>доц. Гуліна І.Г.</i>			

Дніпро
2023

Міністерство освіти і науки України
НТУ «Дніпровська політехніка»

ЗАТВЕРДЖЕНО:
завідувач кафедри
програмного забезпечення комп'ютерних систем
(повна назва)
_____ І.М. Удовик _____
(підпис) (прізвище, ініціали)
« _____ » _____ 2023 року

ЗАВДАННЯ
на кваліфікаційну роботу
бакалавра
(назва освітньо-кваліфікаційного рівня)

студента 122-19-4 Скітенко Анастасії Юрївни
(група) (прізвище та ініціали)
тема кваліфікаційної роботи Розробка веб-орієнтованого застосунку для астрономічної обсерваторії засобами мов HTML, CSS, JavaScript та PHP

затверджена наказом ректора НТУ «ДП» від 16.05.2023 № 350-С

Розділ	Зміст виконання	Термін виконання
Спеціальний	На основі матеріалів виробничої практики та інших науково-технічних джерел провести аналіз стану рішення проблеми та постановку задачі. Обґрунтувати вибір та здійснити реалізацію методів вирішення проблеми	13.05.2023 р.
Економічний	Провести розрахунок трудомісткості розробки програмного забезпечення, витрат на створення ПЗ й тривалості його розробки	27.05.2023 р.

Завдання видав _____ доц. Спірінцев В.В
(підпис) (посада, прізвище, ініціали)
Завдання прийняв до виконання _____ Скітенко А.Ю.
(підпис) (прізвище, ініціали)

Дата видачі завдання: 14.01.2023 р.

Термін подання кваліфікаційної роботи до ЕК: 11.06.2021 р.

РЕФЕРАТ

Пояснювальна записка: 91 с., 52 рис., 6 таб., 3 дод., 21 джерел.

Об'єкт розробки: веб-орієнтований застосунок система для надання послуг дистанційно, розширення можливостей астрономічної обсерваторії та набуття нових відвідувачів.

Мета кваліфікаційної роботи: розробка веб-орієнтованого застосунку для астрономічної обсерваторії засобами мов HTML, CSS, JavaScript та PHP.

У вступі розглядається аналіз та сучасний стан проблеми, конкретизується мета кваліфікаційної роботи та галузь її застосування, наведено обґрунтування актуальності теми та робиться уточнення постановки завдання.

У першому розділі було здійснено аналіз предметної галузі, визначено призначення розробки та актуальність завдання, сформульовано постановку завдання, зазначено вимоги до програмної реалізації, технологій та програмних засобів.

У другому розділі проаналізовані наявні рішення, обрано платформу для розробки, виконано проектування і розробку веб-орієнтованого застосунку, описана його робота, алгоритм і структура його функціонування, а також виклик та завантаження додатку, визначено вхідні і вихідні дані, охарактеризовано склад параметрів технічних засобів.

В економічному розділі визначено трудомісткість розробленої інформаційної системи, проведений підрахунок вартості роботи по створенню додатку та розраховано час на його створення.

Практичне значення полягає у створенні веб-орієнтованої інформаційної системи, що забезпечує: наявність клієнтської частини для користування існуючим контентом та його доповненням через наявність частини адміністратора сайту, та надаватиме можливість обсерваторії легко реалізувати свої товари та послуги.

Актуальність розробки інформаційної системи для автоматизації діяльності компанії в сфері електронної комерції не викликає сумніву та визначається швидко зростаючими тенденціями розвитку космічних інновацій, загальної зацікавленості громадськості у темі космічного простору та масової популяризації даної теми, також актуальність цієї теми обумовлюється розвитком наукового знання в астрономічній галузі

Список ключових слів: ВЕБ-ОРІЄНТОВАНИЙ ЗАСТОСУНОК, АСТРОНОМІЧНА ОБЕРВАТОРІЯ, БАЗА ДАНИХ, PHP, HTML, CSS, JAVASCRIPT.

ABSTRACT

Explanatory note: 91 p., 52 figures, 6 tables, 3 extra, 21 sources.

Object of development: a web-oriented application system for providing services remotely, expanding the capabilities of the astronomical observatory and acquiring new visitors.

The purpose of the qualification work: development of a web-oriented application for an astronomical observatory using the HTML, CSS, JavaScript and PHP languages.

In the introduction, the analysis and current state of the problem is considered, the purpose of the qualification work and the field of its application are specified, the justification of the relevance of the topic is given, and the statement of the task is clarified.

In the first section, the analysis of the subject area was carried out, the purpose of the development and the relevance of the task were determined, the statement of the task was formulated, and the requirements for software implementation, technologies and software tools were specified.

In the second section, available solutions are analyzed, a platform for development is chosen, the design and development of a web-oriented application is performed, its operation, algorithm and structure of its operation are described, as well as application calling and loading, input and output data are determined, and the composition of technical means parameters is characterized.

In the economic section, the labor intensity of the developed information system is determined, the cost of work on creating the application is calculated, and the time for its creation is calculated.

The practical significance lies in the creation of a web-oriented information system that provides: the presence of a client part for using existing content and its addition through the presence of a site administrator part, and will provide an opportunity for the observatory to easily sell its goods and services.

The relevance of the development of an information system for the automation of the company's activities in the field of e-commerce is beyond doubt and is determined by the rapidly growing trends in the development of space innovations, general public interest in the topic of outer space and mass popularization of this topic, and the relevance of this topic is determined by the development of scientific knowledge in the field of astronomy

List of keywords: WEB ORIENTED APPLICATION, ASTRONOMICAL OBSERVATORY, DATABASE, PHP, HTML, CSS, JAVASCRIPT.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

W3C – World Wide Web Consortium;

CMS – Content Management System;

JSON – JavaScript Object Notation;

API – Application Programming Interface;

СУБД – Система управління базами даних;

Sass – Syntactically Awesome Stylesheets;

GUI – Graphical user interface;

MD5 – Message Digest 5;

JS – JavaScript;

SQL – Structured Query Language.

ЗМІСТ

РЕФЕРАТ	3
ABSTRACT	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	5
ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ	10
1.1. Загальні відомості з предметної галузі	10
1.1.1. Основні поняття з теми веб-застосунків.....	10
1.1.2. Технології та принципи розробки веб-сайтів.....	11
1.1.3. Класифікація веб-сайтів за функціональною спрямованістю	16
1.1.4. Висновки за проведеним аналізом	17
1.2. Призначення розробки та галузь застосування.....	18
1.3. Підстава для розробки	21
1.4. Постановка завдання.....	22
1.5. Вимоги до програми або програмного виробу.....	23
1.5.1. Вимоги до функціональних характеристик.....	23
1.5.2. Вимоги до інформаційної безпеки	24
1.5.3. Вимоги до складу та параметрів технічних засобів	24
1.5.4. Вимоги до інформаційної та програмної сумісності.....	25
РОЗДІЛ 2. ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ	26
2.1. Функціональне призначення системи	26
2.2. Опис застосованих математичних методів.....	27
2.3. Опис використаних технологій та мов програмування.....	28
2.4. Опис структури системи та алгоритмів її функціонування.....	31
2.5. Обґрунтування та організація вхідних та вихідних даних програми	43

2.6. Опис розробленої системи	43
2.6.1. Використані технічні засоби	43
2.6.2. Використані програмні засоби.....	44
2.6.3. Виклик та завантаження програми.....	48
2.6.4. Опис інтерфейсу користувача.....	49
РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ	63
3.1. Розрахунок трудомісткості та вартості розробки інформаційної системи ..	63
3.2. Розрахунок витрат на створення інформаційної системи.....	67
ВИСНОВКИ.....	70
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	72
ДОДАТОК А. ЛІСТИНГ ПРОГРАМИ.....	74
ДОДАТОК Б. ВІДГУК КЕРІВНИКА ЕКОНОМІЧНОГО РОЗДІЛУ	90
ДОДАТОК В. ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ	91

ВСТУП

У сучасному світі з кожним днем все більшого впливу набуває інтернет та його ресурси, кількість користувачів та веб-сайтів невідомо зростає, та зростає і попит на надання послуг шляхом інтернету. Наявність веб-сайту вже є обов'язковою для будь-якого підприємства та установи, адже це легкий шлях дізнатися актуальну інформацію, яка потрібна клієнту.

Веб-сайт – незамінний ресурс для швидкого отримання інформації, що цікавить користувачів і клієнтів, що дозволяє не тільки приносити користь у вигляді реклами, але ще й у вигляді спрощення роботи обслуговуючого персоналу і працівників в цілому. Також плюсом є можливість оперативного додавання новин чи коригування вже існуючої інформації, послуг тощо.

З появою нових технологій, що дозволяли підкорювати простір за навколоромною орбітою, вже майже сторіччя як з кожним роком дедалі більше зростає актуальність інформації, що пов'язана з космосом, зростає швидкість появи нових космічних інновацій, тенденції розвитку цієї галузі набирають все більших оборотів. Саме це характеризує попит на розробку інформаційної системи у сферах, що так чи інакше пов'язані із космосом, що б допомагала людям ближче познайомитися з темою космічного простору за рахунок наявного контенту та функціоналу.

Об'єктом дослідження є веб-орієнтований додаток для забезпечення автоматизації діяльності та надання послуг астрономічною обсерваторією, таких як:

- можливість користувачем бронювання та купівлі квитків на поточні події (екскурсії, лекції тощо), що наявні у обсерваторії;
- надання інтерактивно-навчального контенту на тему планет Сонячної системи, та перегляд найвідоміших сузір'їв зоряної карти неба;
- надання інформації про розвиток та дослідження космічного простору, інформацію щодо подій обсерваторії та інше у вигляді наукових та авторських статей.

Таким чином метою даної кваліфікаційної роботи є: розроблення веб-орієнтованого застосунку для астрономічної обсерваторії для надання інтерактивно-навчального контенту на тему космічного простору, можливості дистанційного перегляду наявних екскурсії та бронювання або купівлю квитків на ці заходи, та перегляд і коментування наявних на сайті статей на тему всесвіту та його об'єктів.

Для вирішення наявної мети необхідно буде вирішити наступні задачі:

- здійснити аналіз предметної галузі, опис загальних та технічних відомостей;
- визначення призначення розробки, функціональних характеристик тощо;
- розробити макет дизайну для веб-сайту;
- зробити верстку за наявними даними, такими як: макет дизайну, контент та статті, зображення;
- зробити розробку серверної частини (впровадження усіх функцій за постановкою задачі);
- здійснити загальний опис застосованих методів, архітектур, шаблонів проекту, технологій тощо;
- розрахувати економічну частину, тобто трудомісткості та вартості розробки продукту;
- зробити висновки за виконаною роботою;

Практичне значення полягає у створенні веб-орієнтованого застосунку для астрономічної обсерваторії, що забезпечує наявність клієнтської частини для користування існуючим контентом та його доповненням через наявність частини адміністратора сайту, та надаватиме можливість обсерваторії легко реалізувати свої товари та послуги.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ

1.1. Загальні відомості з предметної галузі

1.1.1. Основні поняття з теми веб-застосунків

Програма-клієнт, що забезпечує читання та відображення інформаційних ресурсів, вмісту веб-документів тощо - зветься веб-браузер або браузер [1]. Також саме він забезпечує передачу даних внесених користувачем до сервера.

Для коректного відображення сайту будь-яким браузером будь-якого виробника наявні впровадженні W3C міжнародні стандарти та рекомендації відносно обробки та відображення даних.

Інформаційні ресурси, що зчитуються браузером представляють собою набір веб-сторінок, що утворюють собою веб-сайт та мають спільну тематику, дизайн, посилання, навігацію тощо. Таким чином веб-сайт це є інформаційна сторінка, доступна для перегляду користувачу за допомогою використання ресурсу інтернет. У свою чергу веб-сторінка це веб-документ, що було розроблено у відповідному форматі гіпертекстової розмітки, що має особисте посилання та може бути відображена веб-браузером.

У кожного веб-сайту може бути різноманітна структура або логічна схема сайту, тобто систематизація сукупності усіх його розділів, підрозділів, пов'язаних сторінок тощо. Усі розділи та підрозділи групуються у деяку ієрархію, наприклад за якимись групами, де виділяються додаткові підгрупи.

Групування починається з головної сторінки веб-сайту, сторінки що є основною та відкривається при вводі посилання до веб-сайту. Надалі за допомогою навігації гіперпосиланнями можна перейти з головної сторінки до внутрішніх сторінок, що будуть відображати контент розділів або підрозділів веб-сайту.

На кожній веб-сторінці сайту користувач може знайти контент або вміст, що було розміщено на сайті. Наприклад графічний контент, текстовий контент, мультимедійні файли тощо. Дизайн сайту зазвичай не відносять до змістової наповненості.

Крім змісту, кожна веб-сторінка має гіперпосилання, текстового або графічного типу, що використовується для переходу до іншої сторінки. Вони можуть бути внутрішніми, тобто забезпечувати перехід на об'єкт наявний на поточному веб-сайті (інша сторінка, файл), та зовнішні, що забезпечують перехід на сторінки або файли іншого сайту.

1.1.2. Технології та принципи розробки веб-сайтів

Характеристика принципів проектування. У процесі створення сайту виділяють розробку внутрішньої частини сайту, тобто розробку основного функціоналу, та розробку зовнішньої частини сайту, тобто верстку сайту.

Створення внутрішньої частини сайту ще називається розробка серверної частини, тобто це розробка програми, що забезпечуватиме роботу сайту шляхом моделі «клієнт-сервер» для успішної взаємодії користувача та самого сервера веб-сайту.

Створення зовнішньої частини сайту це верстка веб-сторінки – процес формування сторінки через такі засоби як: HTML, JavaScript, CSS, PHP та інших. Верстка робиться відповідно до макету дизайну сайту.

Проектування сайтів відштовхується від структури веб-сайту, що застосовується в розробці, тобто його організації та ієрархії потокових веб-сторінок. Логічна схема має бути інтуїтивно зрозумілою кожному користувачеві, що досягається хорошим структуруванням та дизайном. Є кілька видів типів структури:

– лінійна організація, тобто розташування всієї інформації у вигляді лінійного ланцюжка компонентів. Можливість переміщення обмежена, можна перейти лише вперед та назад;

– алфавітна організація, де вся потокова інформація відсортована та виведена в алфавітному порядку;

– деревоподібна організація є найбільш затребуваною і стандартною моделлю, що має на увазі під собою наявність головної сторінки, на якій виведені розділи та підрозділи;

– гібридна організація, тобто комбінація кількох і більше типів схем.

Усі ці структури є варіантами моделі навігації сайту, яка часто розміщується в шапці, або меню сайту [3]. Навігація є одним із головних елементів кожного веб-сайту, бо чим вона прозоріша для користувача, тим легше йому користуватися самим веб-сайтом.

Контент з розділів, що виводиться на екран, відображається в головній частині сторінки або «тілі сайту», під цією частиною знаходиться підвал або футер [3], де найчастіше відображається контактна інформація та угоди користувача.

Увесь контент може бути трьох типів: інтерактивний, динамічний та статичний. Інтерактивними елементами вважаються ті, що будь-яким чином реагують на дії користувачів, наприклад це може бути форма для заповнення, посилання або кнопки. Динамічний контент це та інформація, що може з часом корегуватися, наприклад заголовки, банери тощо. Статичні елементи сайту – незмінний контент, що публікується лише один раз, наприклад розділ допомоги або «запитання – відповідь».

Головні принципи проектування веб-ресурсів:

– дотримання концепції сайту з технічного завдання в дизайні та контенті сайту;

– простота використання інтерфейсу - зрозумілий інтуїтивно дизайн сайту, меню, шапка, футер та елементи навігації на "тілі сайту";

– зручна навігація - легке та зрозуміле здійснення переходу від розділів до підрозділів та навпаки;

– правильна візуальна ієрархія - правильне розташування та використання елементів дизайну, таких як колір, шрифт, текстура тощо - для встановлення

необхідних акцентів, щоб користувач ненавмисно встановлював контакт з контентом, що необхідно було виділити;

- наповнення якісним контентом - весь зміст сайту має підходити під його тематику та бути досить унікальним, корисним та перевіреним;

- швидкодія або швидкий відгук - всі елементи сайту повинні повною мірою відгукуватися на дії користувача, здебільшого це стосується інтерактивного контенту, тобто відгук кнопки натискання, завантаження форми, коректний і швидкий перехід по посиланнях тощо;

- адаптивність (можливість зміни мов, підтримка екранів різної роздільної здатності) - сайт повинен бути гнучким по відношенню до роздільної здатності екрану, оскільки більшість відвідувачів заходить з мобільних пристроїв, так само важливо, щоб був вибір кількох мов для більшої зацікавленості іноземної аудиторії.

Етапи проектування. Першим етапом у розробці є визначення мети та тематики сайту, для цього необхідно визначити напрямок та сферу діяльності сайту, вивчити та оцінити аудиторію, сформулювати цілі створення сайту тощо. Після нарисів початкового плану можна переходити до другого етапу.

Другим етапом буде створення технічного завдання, тобто письмовий опис усіх характеристик, вимог та цілей сайту. Зазвичай до нього входять такі пункти як: 1) мета створення сайту та його призначення; 2) основні вимоги до дизайн-макету; 3) основні системні вимоги; 4) вимоги до функціоналу веб-сайту.

Після створення та доопрацювання технічного завдання можна переходити до третього етапу, тобто розробки прототипів або макетів. На цьому етапі відбувається втілення технічного завдання в макет чорнового вигляду для внесення наступних правок, якщо такі потрібні.

Після доопрацювання всіх прототипів слідує четвертий етап - створення контенту. На цьому етапі прописуються тексти всіх розроблених веб-сторінок, підбираються зображення, відео та інший контент.

П'ятим етапом буде безпосередньо повний дизайн проекту на основі вже готових прототипів та контенту. Після затвердження макету дизайну,

починається етап верстки та адаптації сайту. Даний етап є комплексний, бо потребує опрацювання багатої кількості пунктів, наприклад: сама верстка сайту за допомогою HTML та CSS, тестування кожного інтерактивного елементу (форм, кнопок, посилань тощо), тестування адаптованих розмірів сторінок до екранів різних пристроїв, тестування з використанням різних браузерів тощо.

Зазвичай останнім етапом проектування є наповнення сайту контентом та його публікація.

Класифікація засобів розробки та створення веб-сайтів. Існують варіанти створення сайту через спеціальні сервіси, конструктори, їх перевагою є те, що для їх використання не потрібні знання будь-яких мов програмування, що беруть участь у створенні сайту. Більш розширеними версіями є CMS з функціями конструктора, крім функції налаштування зовнішнього вигляду такі інформаційні системи мають ще можливість додавання та редагування вмісту створеного сайту.

Більш стандартним способом створення сайту є використання мови гіпертекстової розмітки тексту, HTML, що так і розшифровується HyperText Markup Language («мова гіпертекстової розмітки») [1]. Всі елементи цієї мови дозволяють структурувати різні об'єкти, надавати їм оформлення та виводити на екран за допомогою тегів, що записуються у кутові дужки. Код HTML можна доповнювати кодом іншими мовами програмування, наприклад JavaScript або PHP, або використовувати засоби опису зовнішнього вигляду та макета сторінки CSS тощо.

Для отримання доступу до вмісту, написаного HTML, XHTML або XML використовується DOM, «об'єктна модель документа», так само даний програмний інтерфейс дозволяє змінювати структуру, вміст та оформлення таких документів. На даний момент різні браузерні двигуни підтримують стандарти DOM на різних рівнях відповідності.

Для розширення можливостей в оформленні веб-сторінок було розроблено мову таблиць каскадних стилів, а саме CSS. Головною метою його розроблення є відокремлення основного вмісту, написаного мовою розмітки, від оформлення,

що йде до цього вмісту, це дає спрощення оформлення зовнішнього вигляду документа, більшу гнучкість та спрощення читання документу.

Однією з популярних мов програмування, що призначена для створення сайтів, є PHP. Принцип його роботи полягає в тому, що скрипти, написані цією мовою, виконуються на сервері і тільки після результату роботи передаються користувачеві. PHP – є впроваджуваною мовою сценаріїв, що дозволяє швидко створювати динамічні сторінки з інтерактивним наповненням, наприклад, обробка даних форм, виведення повідомлень тощо. Застосовувати цю мову можна як у створенні сайтів з модулів, так і у створенні невеликих програм.

JavaScript також використовується як мова сценаріїв для створення інтерактивності веб-сторінки, проте принцип дії у нього інший, такий код передається на клієнтську машину і виконується вже на ній. Ця мова програмування має широкі можливості взаємодії з елементами сторінки без перезавантаження самої сторінки, що дає велику варіативність дій при написанні скриптів.

DHTML або Dynamic HTML – це підхід створення інтерактивного веб-сайту, що поєднує в собі основні чотири технології, такі як: мови розмітки HTML, каскадних таблиць стилів CSS, скриптової мови JavaScript та стандарти DOM.

Для роботи з інформацією на веб-сайтах часто використовують бази даних різних видів, що допомагають не тільки зберігати інформацію, але ще й користуватися пошуком серед присутньої інформації, сортувати її та виводити результати пошуку за певними параметрами. Бази даних поділяються на реляційні (наприклад Oracle DB, MySQL, PostgreSQL тощо), що зберігають інформацію у вигляді таблиць та строк, та нереляційні (наприклад MongoDB, Couchbase тощо), що розміщують та зберігають інформацію у колекційних документах JSON.

1.1.3. Класифікація веб-сайтів за функціональною спрямованістю

Загальноприйнятої класифікації сайтів на даний час не існує, однак їх можна групувати за їх спрямованістю, наприклад комерційні (сайти послуг, інтернет-магазини тощо), інформаційні (блоги, новини), трафікові (онлайн-кінотеатри, довідники тощо), соціальні (соцмережі, форуми, портали); та за функціями, які вони надають користувачам, що є найпоширенішою класифікацією.

То ж за функціональним спрямуванням розрізняють такі типи веб-сайтів:

Односторінковий сайт. Даний сайт є однією чи декількома сторінками, що розроблені для комерційних цілей, може мати різну довжину (як короткі сторінки без скролінгу (прокручування), так і довгі схожі на слайд-шоу сайти). Їх ще називають посадкові сторінки (Landing Page), в основному створюються для реклами будь-якого вже відомого продукту або новинки на ринку, збільшення потенційних клієнтів та проведення аналізу впливу на цільову аудиторію маркетингових прийомів, що було використано.

Корпоративний сайт або презентаційні сайти, розроблюються з метою представити компанію, тобто цей вид сайтів є візитівкою або ж «обличчям» компанії, товару, деякого діяча тощо. Характерною рисою даних сайтів є розділи з історією, загальною інформацією, контактними даними та новинами, якщо це компанія або людина - інколи може бути наявний розділ з портфоліо. Розроблюється для підвищення іміджу компанії/діяча/продукту, регулярних надходжень нової інформації щодо діяльності, розміщення контактних даних підприємства.

Онлайн-сервіси – це такі веб-сайти, що призначені для надання послуг різного формату, наприклад електронні пошти, онлайн кінотеатри, онлайн-бібліотеки тощо. В свою чергу також поділяються на різні види за функціональним призначенням, на рис. 1.1 можна побачити чотири основні види онлайн-сервісів:

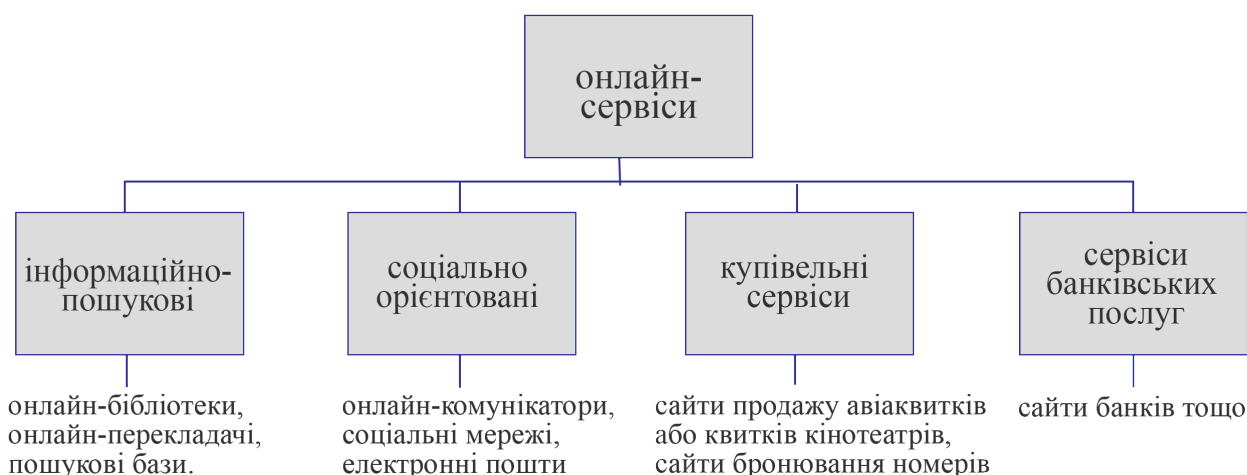


Рис. 1.1. Схема поділу онлайн-сервісів

Інтернет-магазин це онлайн-сервіс продажу товарів, проте на даний момент такого роду сайтів налічується досить багато, що в основному його можна виділити як окрему категорії класифікації веб-сайтів. Головна мета таких сайтів – продаж товарів шляхом каталогу товарів магазину в онлайн режимі. Великі магазини або підприємства часто вводять онлайн-продаж для збільшення відсотка збуту, так як з кожним днем це стає все популярнішим.

Медіа ресурси ще можна назвати як інформаційно-новинні портали, так само це можуть бути особисті блоги з інформацією будь-якого формату, або ж змішаний ресурс і з актуальною інформацією, і з авторським контентом (наприклад думка автора: рецензії, відгуки тощо). Головною метою цих сайтів є інформування, систематизація інформації, її актуальність, швидко та легко доступність.

1.1.4. Висновки за проведеним аналізом

За результатом проведеного вище аналізу можна виділити головні плюси, що задовольняють мету розробки, у кожного з типу сайтів, та надалі адаптувати їх під розроблюваний веб-застосунок для астрономічної обсерваторії.

Отже головним плюсом односторінкового сайту є те, що односторінковий сайт, або лендінг, може бути головною сторінкою для залучення потенційних користувачів красивою картинкою і деяким слайд-шоу, що є рекламою функцій, що є на сайті. Також плюсом є те, що такі сторінки легко адаптуються під мобільну версію сайтів.

Головними плюсами корпоративного сайту, в свою чергу, є те, що на цих сайтах наявна інформація щодо організацій, робітників тощо. Ця інформація подається у вигляді короткої історії створення та розвитку, перелік досягнень та відгуків клієнтів, блог компанії та її співробітників.

У онлайн-сервісах буде розглядатись сервіс продажу квитків, що має схожі принципи дії з онлайн-магазином, тому переваги в них однакові - головним плюсом є реалізація товару дистанційно, можливість дистанційного перегляду наявних подій за датами та перегляд вартості відвідування відповідної події.

У медіа ресурсів, або інформаційно-новинного порталу, головним плюсом є усі функції, що дозволяють будь-яким чином взаємодіяти з інформацією, тобто публікація, редагування або видалення статей адміністратором сайту та можливість перегляду та коментування опублікованих статей у користувача.

Однак для здійснення мети розробки повною мірою необхідно розробити також специфічні функції, які в основному не присутні на сайтах, що були розглянуті. Таким чином, щоб удосконалити отримані плюси та досягти потрібного повною мірою набору функціоналу, буде додано розділ інтерактивного перегляду інформації про відомі сузір'я зоряного неба України та планети Сонячної системи.

1.2. Призначення розробки та галузь застосування

Призначенням розробки є розробка веб-орієнтованого застосунку для астрономічної обсерваторії, у якому буде наявна клієнтська частина та частина адміністрування. Користувач матиме нагоду дистанційно переглядати наявні екскурсії, бронювати та купувати квитки до них, ознайомлюватися з статтями на

тематику космічного простору та сонячної системи. Частина адміністрування буде полягати у додаванні нових записів щодо екскурсій та подій, та додаванні нових статей у розділ новин.

Розроблений продукт буде орієнтовано на відвідувачів астрономічної обсерваторії, для якої розробляється сайт, та усіх людей, що виявляють будь-який інтерес до космічного простору. Аудиторія сайту не може бути аналізована та розсортована за віковими групами, оскільки веб-сайт призначений для всіх вікових груп.

Веб-орієнтований застосунок призначений для:

- дистанційного перегляду наявних подій та екскурсій, що проводяться у астрономічній обсерваторії;
- купівлі або бронювання білетів на наявні заходи та екскурсії реєстрованими користувачами;
- перегляду науково-розважального контенту на тему космічного простору, космічних технології тощо;
- коментування наявних на веб-сайті статей науково-розважального контенту реєстрованими користувачами;
- перегляду інтерактивної інформації щодо популярних сузір'їв зоряного неба України та планет Сонячної системи;
- додавання нових статей у розділ новин та нових записів щодо екскурсій адміністратором сайту.

Головна сторінка розроблюваного веб-застосунку буде зроблена у вигляді лендінг-сторінки, що ознайомлюватиме користувачів з функціоналом сайту через відповідні банери. На рис. 1.2 можна побачити прототип першого фрейму головної сторінки, де наявна шапка з елементами навігації, та посередині буде банер з привітальним текстом для користувачів. Надалі нижче будуть реалізовані два блоки, один з актуальною інформацією та посиланнями на відповідні розділи, де буде знаходитися дана інформація, та блок з тестом. Після усіх блоків з контентом буде знаходитись футер сайту.

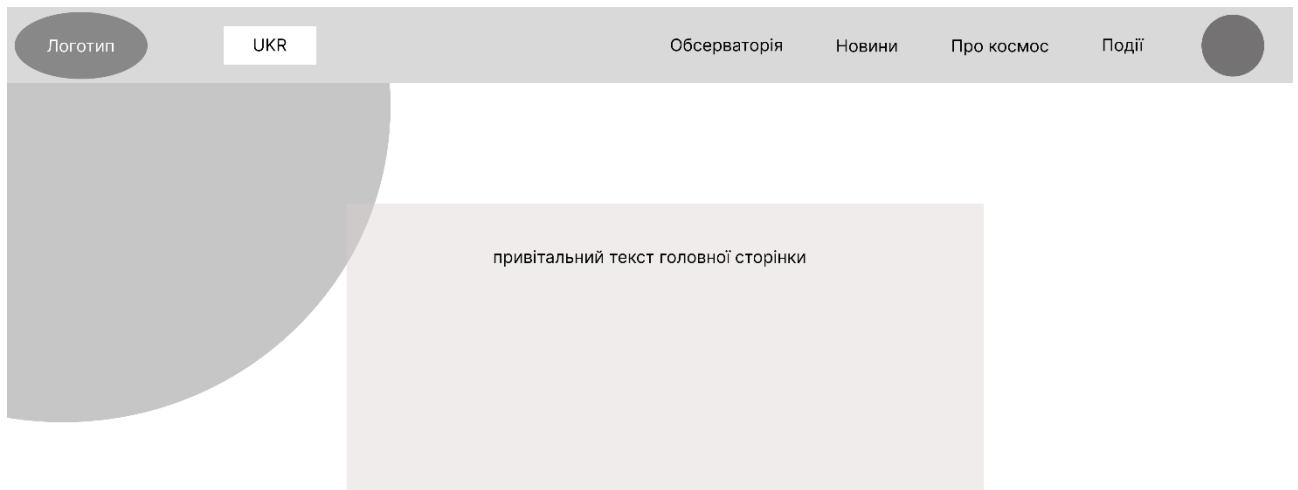


Рис. 1.2. Прототип головної сторінки

На підставі того, що внутрішні сторінки повинні надавати певні функції та виводити певний контент, що є у розділах або підрозділах, прототип макета для таких сторінок відрізняється від макету головної сторінки, та його можна побачити на рисунку рис. 1.3. Шапка сайту та футер остаються незмінними, однак тіло сайту буде містити контейнер для контенту та панель швидкого доступу (меню-бар).

Залежно від розділу веб-сайту може також змінюватися і структура контейнера, наприклад, у розділі "Про космос" контейнер у тілі сайту не міститиме панель швидкого доступу з-за наявності інтерактивно-навчального контенту на цій сторінці, а у розділі «Профіль», за умовою, що користувача було авторизовано, панель швидкого доступу буде складатися з елементів управління профілем користувача, якщо авторизацію здійснено не було, меню-бар буде складатися з стандартного контенту.

Так як за метою розробки та обраними принципами програмування основою дизайну є його адаптивність, було розроблено макети і для мобільної версії сайту. Головна та внутрішні сторінки будуть повторювати повний зміст, однак його розташування буде змінено, що можна побачити з рис. 1.4 а, б.

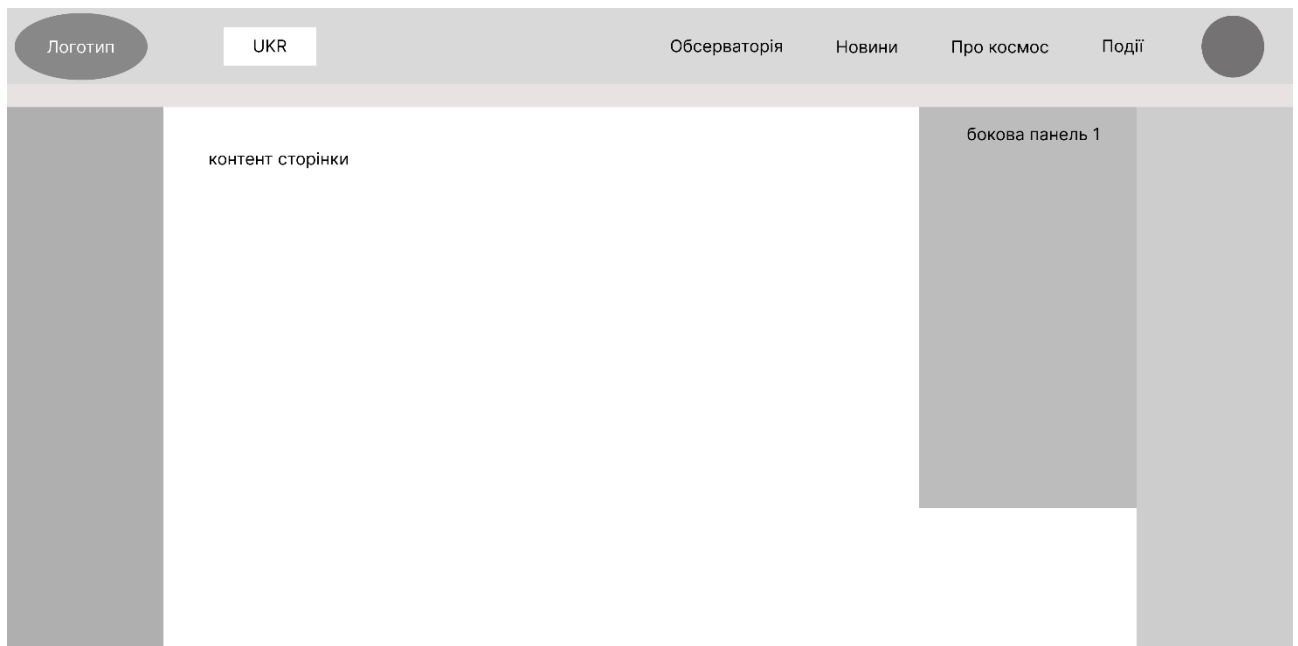


Рис. 1.3. Прототип макету внутрішніх сторінок

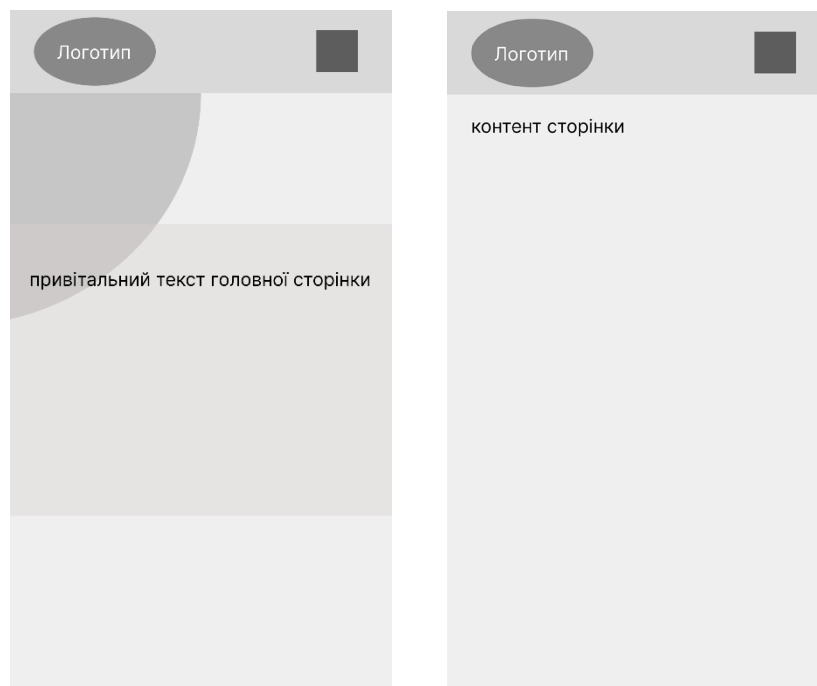


Рис. 1.4 а, б. Прототипи для мобільної версії: а) головна сторінка, б) внутрішня сторінка

1.3. Підстава для розробки

Перелік підстав для виконання кваліфікаційної роботи представлено нижче:

- освітня програма за спеціальністю 122 «Комп’ютерні науки»;
- навчальний план та графік навчального процесу;
- наказ ректора Національного технічного університету «Дніпровська політехніка» № 350-С від 16.05.2023 р;
- завдання на кваліфікаційну роботу за темою «Розробка веб-орієнтованого застосунку для астрономічної обсерваторії засобами HTML, CSS, JavaScript, PHP».

1.4. Постановка завдання

Головною метою даної кваліфікаційної роботи є розробка інформаційної веб-орієнтованої системи для надання послуг дистанційно, розширення можливостей астрономічної обсерваторії та набуття нових відвідувачів.

Призначенням програми є надання користувачам змоги: дистанційно переглядати наявні заходи, екскурсії тощо, бронювати або купувати квитки задля відвідування, переглядати наявні статті на сайті, коментувати їх, переглядати інтерактивно-навчальний контент на тему планет та сузір’їв.

За проведеним вище аналізом предметної області веб-сайт повинен підтримувати такі параметри як:

- багатоплатформність (дизайн та функції повинні бути адаптовані до різної роздільної здатності екрана: комп’ютера, телефону тощо);
- інтуїтивний інтерфейс та навігація (веб-сайт має бути зрозумілим для користувача будь-якої вікової групи);
- підтримувати кілька мов (для припливу іноземної аудиторії, що цікавиться цією сферою, сайт має підтримувати англійську мову);
- інтерактивність та функціонал (сайт повинен відкликатися на дії користувача, мати повний перелік функціоналу, що було представлено тощо).

Та також за проведеним вище аналізом можна виділити основні етапи розробки:

- визначення мети, напрямку та тематики сайту, аналіз аудиторії, формування основних цілей створення сайту;
- створення розгорнутого технічного завдання, тобто детальний опис усіх вимог до дизайну, функціоналу, змісту тощо;
- розробка чорнових макетів дизайну та внесення правок;
- розробка змісту сайту (статті, головний текст, банери, зображення тощо);
- розробка головного макету сайту з використанням чорнового макету та розробленого контенту;
- етап верстки та адаптації сайту (тестування та налагодження);
- наповнення розробки змістом та публікація.

1.5. Вимоги до програми або програмного виробу

1.5.1. Вимоги до функціональних характеристик

За проведеним вище аналізом, головними вимогами щодо функціональних характеристик розроблюваного веб-орієнтованого застосунку є:

- інтуїтивний інтерфейс – легкі на сприйняття користувачем усі елементи присутні на сайті;
- прозора навігація – можливість швидкого здійснення переходу між розділами та підрозділами сайту;
- мультимовність – підтримка українського та англійської мов;
- швидкодія (швидкий відгук) – функціонал сайту повинен відкликатися на усі дії користувача за прийнятний час завантаження;
- адаптивність дизайну – підтримка екранів різної роздільної здатності, таких як комп'ютер, планшет та телефон;
- конфіденційність даних – усі особисті дані будуть захищені особистим кодованим паролем профіля користувача.

1.5.2. Вимоги до інформаційної безпеки

Інформаційна безпека – це стан захищеності інформації та самої інформаційної системи від несанкціонованого доступу, та будь-якого впливу на неї [1]. Впровадження способів захисту це першорядна мета при розробці будь-якого виду веб-орієнтованого додатку, щоб уникнути проблем з доступом та збереженням інформації в розробці буде реалізовано наступні пункти:

- конфіденційність особистої інформації (шляхом створення окремих облікових записів захищених кодованим паролем у форматі MD5, інформація в обліковому записі доступна тільки користувачу, хто є власником даного профілю);

- цілісність інформації, що зберігається (зберігання інформації буде здійснюватися зі збереженням структури та змісту при операціях передачі, отримання та зберігання даної інформації);

- доступність інформації (інформація на запит користувача буде отримуватись за прийнятний час завантаження);

- використання ліцензійного та перевіреного програмного забезпечення;

- безпечне підключення (збереження особистої інформації, паролів та інше будуть захищені).

1.5.3. Вимоги до складу та параметрів технічних засобів

Для використання даної інформаційної системи не потрібна наявність специфічного комп'ютерного або мобільного обладнання, достатньо середніх мінімальних вимог таких як:

Для комп'ютерів:

- двоядерний процесор Intel Celeron 1,5 GHz та вище;

- оперативна пам'ять 4 Гб та вище;

- наявність миші, клавіатури та монітора;

- наявність підключення до ресурсу «інтернет»;

- наявність браузера будь-якого типу.

Для телефонів та планшетів:

- графічний процесор від 700 МГц;
- процесор Octacore із частотою 1,4 GHz та вище;
- OpenGL версія 3+;
- наявність підключення до ресурсу «Інтернет»;
- наявність браузера будь-якого типу.

1.5.4. Вимоги до інформаційної та програмної сумісності

Для програмної сумісності потрібно наявність на пристрої мережевої карти та останньої версії мережевого драйверу до неї. Головною вимогою є підключення до ресурсу «Інтернет» з стабільною швидкістю від 3 Мбіт/с для мобільного інтернету та від 20 Мбіт/с для мереж на комп'ютерах, та наявність будь-якого браузера.

Вимогами до програмного забезпечення пристроїв є:

- для комп'ютерів - операційна система Microsoft Windows 7/8.1/10/11 (32 або 64 bit);
- для мобільних пристроїв: андроїд версії 7 та вище.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1. Функціональне призначення системи

Головною метою кваліфікаційної роботи є розробка веб-орієнтованого застосунку для астрономічної обсерваторії для надання інтерактивно-навчального контенту на тему космічного простору, можливості дистанційного перегляду наявних екскурсій та бронювання або купівлю квитків на ці заходи. Також на цьому веб-сайті буде наявний розділ з науковими статтями, де кожен зареєстрований користувач має нагоду залишити свій коментар.

На сайті наявні два рівні доступу, що надають право користування різними функціями, що наявні на сайті. Повним функціоналом на рівні «Користувач» є:

- реєстрація профілю користувача, подальший логін у нього, редагування та видалення;

- перегляд наявного на веб-сайті контенту;
- бронювання та купівля квитків;
- можливість залишення коментарів під статтями;

Повним функціоналом на рівні «Адміністратор» є:

- додавання нової інформації щодо заходів;
- додавання нових статей.

Основними функціями, що повинен виконувати розроблюваний веб-орієнтований застосунок є:

- адаптація інтерфейсу користувача відносно роздільної здатності екрану пристроїв, таких як комп'ютер, телефон тощо;
- змінення мови інтерфейсу (підтримка української та англійської);
- реєстрація, редагування та видалення профілю користувача;

- бронювання та купівля квитків;
- перегляд інтерактивно-навчального контент на тему планет сонячної системи та сузір'їв нічного неба України;
- перегляд та коментування статей, що було опубліковано на сайті;
- додавання нових записів до розділів «новини» та «події» адміністратором сайту;
- надання інформації про розвиток та дослідження космічного простору.

Для реалізації усього переліченого вище функціоналу на створюваному додатку потрібно:

- розробити дизайн для різної роздільної здатності екрану;
- реалізувати можливість перекладення змісту сайту на англійську мову;
- реалізувати базу даних для збереження поточної інформації;
- реалізувати за допомогою веб-форми такі функції для користувачів як: реєстрація профілю користувача, подальше редагування, видалення та вихід та вхід у профіль, додавання заброньованих квитків у профіль зареєстрованого користувача, подальша оплата, додавання зареєстрованим користувачем коментарів до опублікованих статей;
- реалізувати за допомогою веб-форми такі функції для адміністратора, як: додавання нових публікацій у розділ «новини», додавання нових записів щодо заходів до розділу «події»;
- реалізація сторінки, що буде містити інтерактивний контент щодо найвідоміших сузір'їв зоряного неба та планет Сонячної системи з використанням 3D моделей та анімації;

2.2. Опис застосованих математичних методів

При реалізації завдання кваліфікаційної роботи не застосовувалися будь-які математичні методи, окрім мінімальних базових арифметичних дій за потребою, так як властивості предметної області не передбачають використання складніших математичних методів.

2.3. Опис використаних технологій та мов програмування

Для розробки усіх макетів був обраний онлайн-сервіс Figma, що надає можливість розробляти елементи інтерфейсу, інтерактивні прототипи та легке конвертування зроблених макетів у різні формати, наприклад pdf, png тощо.

Переваги обраного графічного багатоплатформного онлайн-редактора перед його іншими аналогами в тому, що:

- є стандартом у промисловості розробки;
- даним сервісом можна скористатися через браузер;
- зберігання файлів у хмарному зберіганні.

Для верстки сайту була обрана версія HTML5, це остання версія, що широко використовується для розробки сьогоденних веб-орієнтованих застосунків та не тільки. Перевагами HTML5 між аналогами мов та більш ранніх версій є:

- спрощена та скорочена кількість використовуваних тегів для розмітки;
- поліпшений функціонал, наприклад, можливість використання віртуальної векторної графіки;
- зберігання даних у кеші програми, що дозволяє серверної частини написаної JavaScript працювати у фоновому режимі;
- легкодоступна документація та безліч різних прикладів реалізації того чи іншого завдання;

Для оформлення сторінок була використана мова CSS3, якою легко можна надавати будь-який бажаний зовнішній вигляд як самій сторінці браузера, так і інтерактивним елементам на ній. Головними перевагами використання каскадних таблиць стилів є:

- простота написання, використання та впровадження;
- зменшення кількості коду мовою HTML, відповідно швидше завантаження сторінок;
- широкий набір функціоналу щодо оформлення;
- надає можливість адаптації дизайну під різні пристрої та роздільну

здатність екранів.

Для спрощення та прискорення процесу написання CSS-коду була використана метамова Sass, що є передпроцесором мови CSS [6]. Sass - має свій особистий синтаксис та розширення файлів .scss, з якого генерується стандартні файли розширення .css. Має багато переваг, одні з яких представлені нижче:

- дає можливість більш простого та швидкого оновлення та масштабування файлів мови CSS;

- дає можливість використовувати змінні різних типів для оголошення: рядків, списків, числових значень, параметрів кольору, логічних значень тощо.

Для конвертації файлів написаних мовою передпроцесора у стандартний вигляд була використана GUI-додаток Koala з відкритим вихідним кодом.

Головною перевагою даної програми є те, що вона є безкоштовною, постійно оновлюваною і має функцію прослуховувати файли і автоматично їх компілювати при будь-якій зміні файлу, працюючи при цьому фоновно і не вимагаючи жодних дій з боку користувача, так само дозволяє вибрати шлях де буде розміщуватися файл, що компілюється.

Типом архітектури для даної інформаційної системи було обрано клієнт-серверну архітектуру трирівневого типу, схему якої можна побачити на рис. 2.1. У даному типі архітектури виділяють три рівні: 1) сервер баз даних, файловий сервер, 2) сервер програми (бізнес-логіка та логіка даних), 3) клієнтська частина [5]. Усі звернення з клієнта до бази даних відбуваються через проміжне програмне забезпечення, що знаходиться на сервері. Дана організація дозволяє зробити систему більш гнучкою та ефективною та підвищити її продуктивність. Також можна виділити такі переваги, як:

- багаторазовість повторного виконання оброблюваних сервером запитів з клієнта з економією ресурсів системи, що витрачаються на обробку;

- паралельність роботи сервера додатків та сервера бази даних, яка не залежить від потужності та функціоналу сервера додатків;

- забезпечення легкого та швидкого доступу до бази даних з клієнтів шляхом виконання запитів сервером додатків.

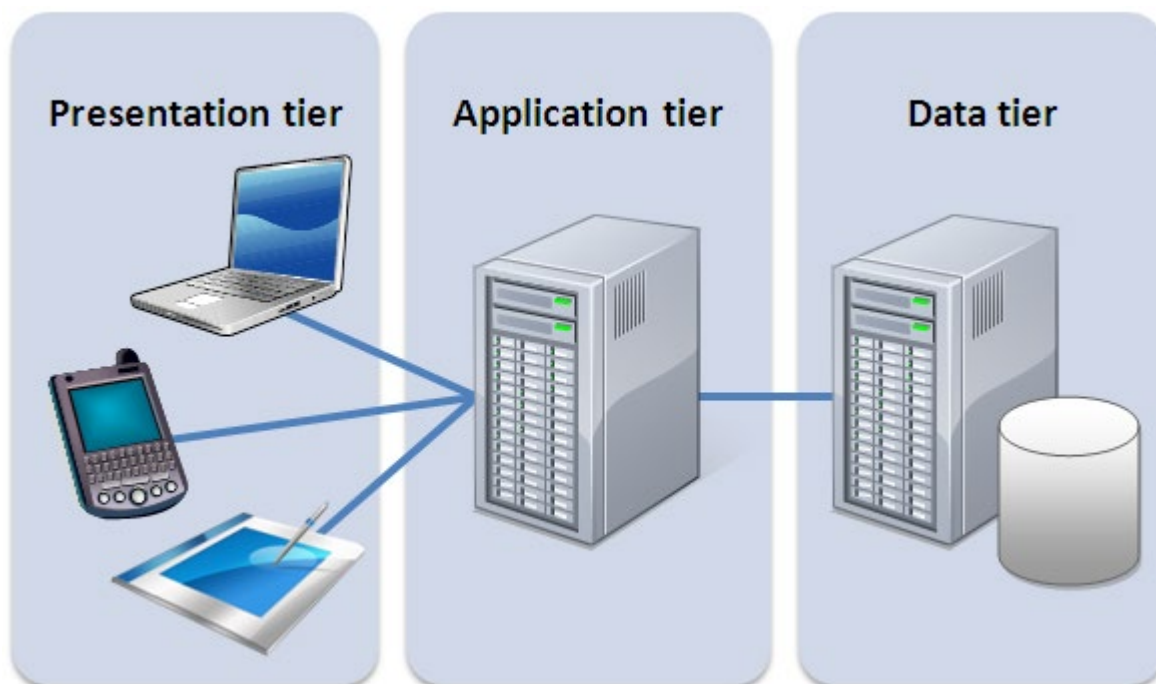


Рис. 2.1. Тривірнева архітектура клієнт-сервер

Мова JavaScript підтримує багато стилів програмування, та є майже універсальною мовою розробки, широко використовується для придання інтерактивності веб-сторінкам. Головними перевагами є:

- підтримується усіма сучасними браузерами;
- легко інтегрується в мову HTML та у серверну частину додатку;
- дозволяє прискорити роботу продукту, що розробляється;
- має багату кількість корисних фреймворків та допоміжних бібліотек.

Для реалізації інтерактивного контенту у вигляді перегляду короткої інформації про планети Сонячної системи було обрано бібліотеку `three.js`. Кросбраузерна бібліотека мови JavaScript, що використовується в розробці веб-додатків для моделювання та відображення анімованої комп'ютерної 3D графіки.

Завдяки WebGL, низькорівневому графічному API, `Three.js` дозволяє створювати 3D-анімацію з прискоренням графічного процесора, без необхідності використання пропрієтарних плагінів для браузера [7]. Має багатий функціонал щодо рендерингу сцени та фігур, додавання текстур тощо, не потребує окремого створення 3D моделей.

Обраною базою даних для управління даними у розробці веб-

орієнтованого застосунку було обрано реляційну базу даних MySQL з-за ряду переваг, таких як:

- зрозумілий інтерфейс та легке завантаження;
- великий функціонал та багата кількість додаткових плагінів, що спрощують роботу з СУБД;
- швидке реагування на запити та їх виконання (є однією з найшвидших СУБД на ринку);
- високі показники безпеки (за рахунок шифрування даних користувача тощо) та доступності (за рахунок наявності додаткових вузлів сервера, що застосовуються в аварійних ситуаціях);
- легке масштабованість горизонтально та вертикально.

Як засіб для взаємодії серверу та бази даних було обрано мову програмування php. Ця мова програмування має дві робочі бібліотеки для підключення і роботи з базою даних MySQL, з цих двох бібліотек для роботи з даною базою даної була обрана бібліотека MySQLi через її спрямованість на реалізацію всіх можливостей СУБД, швидку технічну підтримку та оновлення нових версій, і наявність більшої продуктивності та швидкості реалізації запитів. Крім цього, також можна виділяти такі переваги мови php, як:

- гнучкість написання скриптів, простота використання та розуміння синтаксису;
- збільшення стандартного функціоналу за рахунок наявності великої кількості додаткових фреймворків та бібліотек.

2.4. Опис структури системи та алгоритмів її функціонування

В розроблюваному веб-орієнтованому застосунку для астрономічної обсерваторії за тип логічної схеми було взято деревоподібну організацію, таким чином на веб-сайті наявна така структура сторінок:

- розділ "головна сторінка" - навігація здійснюється натисканням на логотип в шапці сайту, є лендинг-сторінкою, що містить у собі 3 фрейми з

банерами з посиланнями на розділи та текстом;

- розділ "Обсерваторія" - містить загальну інформацію про обсерваторію та її працівників, є елементом корпоративної сторінки;

- розділ "Новини" - містить наявні у базі даних статті, при переході за посиланням у назві статті - відкривається підрозділ із самою статтею та наявністю поля коментаря на ній;

- розділ "Про космос" - містить у собі схему п'яти популярних сузір'їв та їх назви, п'ять планет із Сонячної системи, для яких доступний перегляд 3D-моделі та короткої інформації про них;

- розділ "Події" - містить виведені з бази даних про події за різні проміжки часу;

- розділ "Профіль" - перехід здійснюється за іконкою, або за посиланням у списку, що містить, містить веб-форми для входу або реєстрації і подальшим взаємодією з профілем при авторизації користувача.

У директорії проекту (рис. 2.2) застосовується групування робочих файлів за мовою програмування для здійснення швидкого пошуку необхідних елементів, файли написані мовою HTML та PHP містяться у кореневій папці для простоти навігації.

Папки в свою чергу поділяються на кілька функціональних груп: папки з файлами, призначеними для оформлення, папка зі скриптами на мові JavaScript і папка з підключеними до проекту модулями (бібліотеками).

Серед папок призначених для оформлення сторінок є папки безпосередньо з кодом каскадних стилей та його предпроцесором, та папки з шрифтами, що підключаються, і зображеннями, що використовуються у розроблюємому веб-орієнтованому додатку.

Файли, що містить папка `scss` оброблює GUI-додаток для обробки предпроцесорної мови Sass – Koala, та розміщує конвертовані файли у директорії `css` ці директорії можна побачити на рис. 2.3 а, б. Усі файли мовою CSS відповідають за оформлення сторінок, що були написані мовою HTML, призначення кожного наступне:

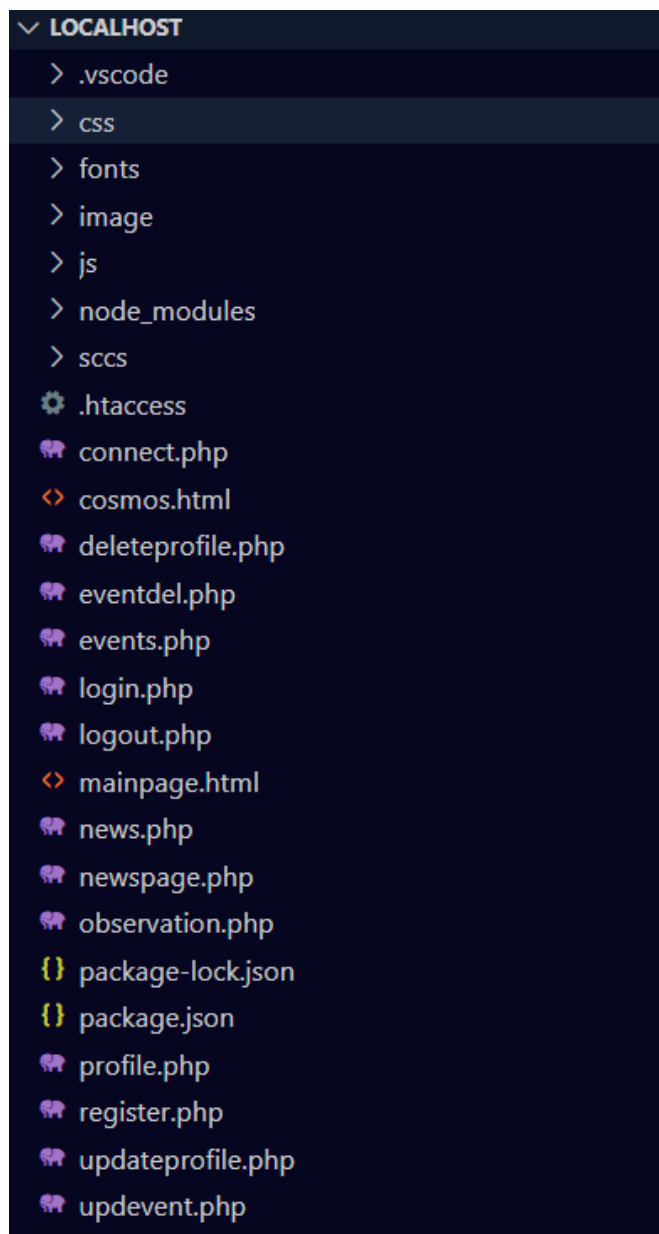


Рис. 2.2. Повна структура директорії

- fonts.css – файл відповідний за підключення усіх наявних у проекті шрифтів;
- form.css – файл відповідний за налаштування форм у розділі профіля користувача;
- reset.css – файл відповідний за «скидання» наявних у браузері стандартних налаштувань стилю;
- style.css – файл з головною частиною оформлення усіх наявних HTML сторінок

– усі файли з розширенням файлу .css.map - «вихідна карта» однойменного файлу – є спеціальним файлом, який з'єднує передпроцесорну версію коду з оригінальною версією (файлами мовою CSS).

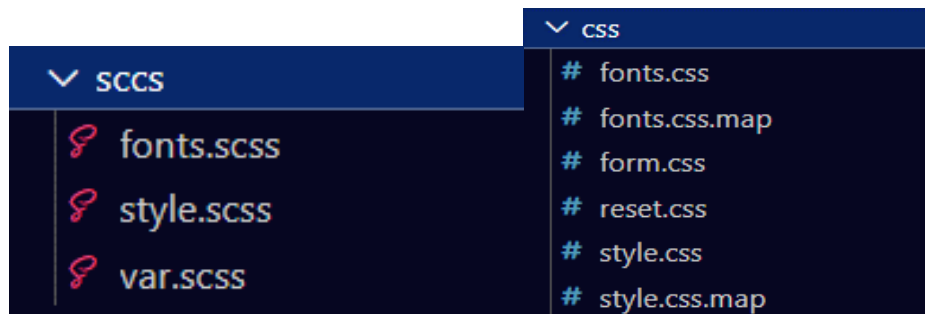


Рис. 2.3 а, б. Папки головної директорії проекту: а) scss, б) css

У ході створення проекту було прийнято рішення підключати шрифти (найменуванням CormorantGaramond та PlayfairDisplaySC звичайного, курсивного та жирних накреслень) та зображення, що повинні бути наявними на розроблюваному сайті, з робочої директорії (фрагмент скріншоту цих папок можна побачити на рис. 2.4 а, б).

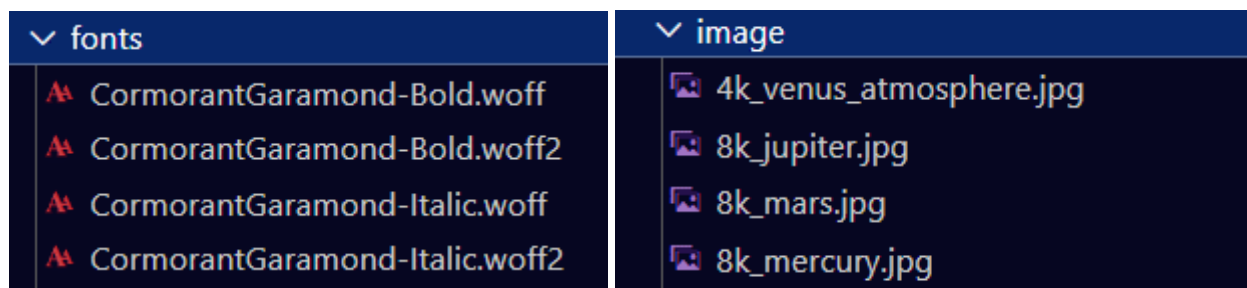


Рис. 2.4 а, б. Фрагменти скріншотів папок: а) fonts, б) image

У директорії js, яку можна побачити на рис. 2.5, зібрані всі присутні в проекті файли написані скриптовою мовою JavaScript:

– 3dscene.js – файл, що відповідний за відрисовку та анімацію 3D-моделей у розділі «Про космос», використовує методи бібліотеки THREE.js;

– clock.js – файл з скриптом, що відповідний за наявність годинника у панелі меню-бар на внутрішніх сторінках сайту;

– dropdownlistbtn.js – містить дві функції для коректної роботи (відкриття або закриття списку при натисканні на кнопку) випадаючого списку з посилань в шапці сайту при зменшенні роздільної здатності екрана нижче 900 пікселів шириною;

– formvalid.js – містить функції для перевірки наявних форм реєстрації та входу до профілю користувача на коректне заповнення усіх даних форми користувачем;

– lang.js – «файл-словник» з масивом, що містить переклад усіх занесених до нього змістових текстів сайту;

– translate.js – файл, що містить скрипт функції, що переводить сторінки сайту шляхом заміщення фраз за спеціальною назвою класу, що містить префікс «lng-»;

– solrsystembanner.js – файл, що відповідний за відрисовку та анімацію 3D-моделей у моделі руху Сонячної системи;

– starbg.js – файл, що відповідний за відрисовку та анімацію останнього банеру з зірками та Луною на головній сторінці;

– suntimes.js – бібліотека відкритого коду з формулами для розрахунку часу сходу та заходу сонця за координатами широти та довготи поточного розташування (вихідним було обрано місто Дніпро) та поточної дати. У ході роботи було доопрацьовано для можливості виведення часу у стандартному 24-годинному вигляді з годинником та хвилинами.

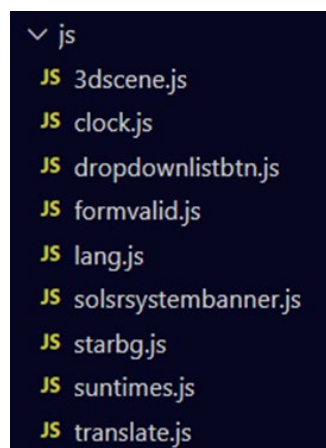


Рис. 2.5. Директорія файлів мови JavaScript

Для подальшої роботи з 3D графікою бібліотека THREE.js була підключена за допомогою менеджера пакетів Node.js і записана у файли мови JSON `package.json`, що розміщені у корневій директорії проекту. Усі підвантажені модулі розміщуються в папці `node_modules`, відповідну директорію та файли мови JSON можна побачити на рис. 2.6 а, б.

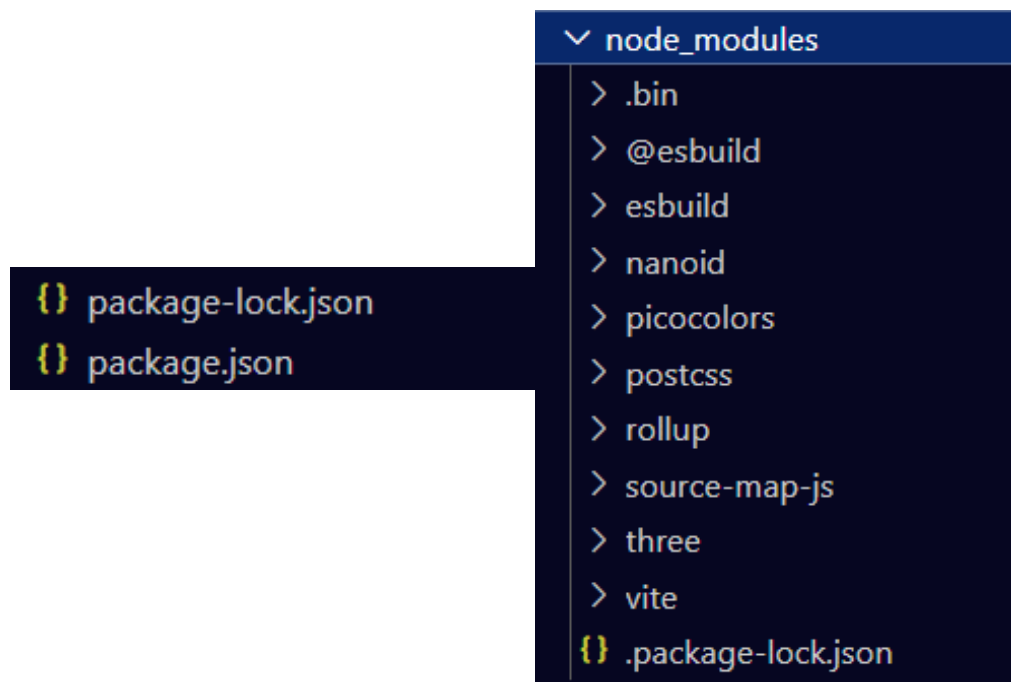


Рис. 2.6 а, б. Підключені модулі: а) файли мови JSON, б) директорія з модулями

Коренева директорія проекту (рис. 2.7) містить файли, що були написані мовами HTML і PHP, тобто самі сторінки сайту та допоміжні файли, що дозволяють взаємодіяти з базою даної.

Відповідність файлів до їх функцій представлена нижче:

- `mainpage.html` - файл відповідає за зміст лендінг сторінки, або основної сторінки сайту;
- `cosmos.html` - файл, що відповідає за зміст розділу "Про космос";
- `connect.php` - допоміжний файл із скриптом підключення до бази даних;
- `profile.php` - головний файл сторінки розділу "Профіль";
- `login.php`, `logout.php`, `register.php`, `updateprofile.php`, `updateevent.php` -

допоміжні файли розділу "Профіль";

– news.php - файл веб-сторінки розділу "Новини";

– newspage.php - сторінка підрозділу розділу "Новини", або сторінка, куди виводиться відкрита користувачем стаття.

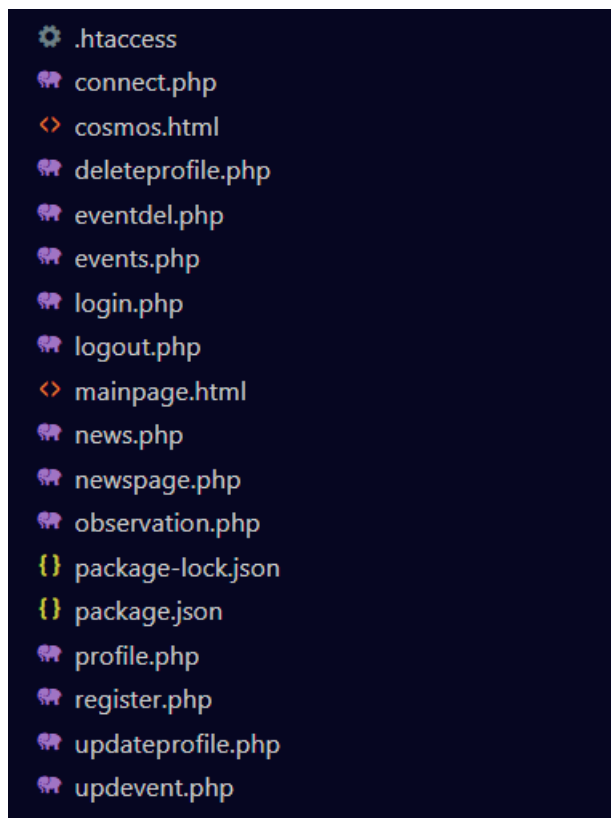


Рис. 2.7. Файли кореневої папки проекту

База даних MySQL є реляційною, тому для кожної сутності необхідна наявність атрибутів, що зберігаються, таким чином потрібно виділити ключові об'єкти, які будуть визначати значення інших елементів даних за допомогою первинних та альтернативних ключів. У базі даних буде шість сутностей: таблиця, що буде відповідати за профіль користувача, таблиця з наявними подіями (або ще можна сказати з інформацією до наявних квитків), таблиця, що буде відповідати за придбання або бронювання квитків до наявних подій, таблиця з наявними на сайті статтями та таблиця з коментарями користувачів до статей, що наявні у попередній таблиці (рис. 2.8). Взаємодія з базою даних відбувається за допомогою скриптів мови PHP, через допоміжний скрипт

підключення до бази даних. Після успішного підключення виконуються різні функції виведення даних із існуючих у базі даних таблиць за допомогою веб-форм та текстових елементів. Для взаємодії користувача з системою та виконання його запитів було обрано комунікацію та надсилання запитів через веб-форми, що присутні на сайті в різних розділах, оскільки форми є інтуїтивно зрозумілими будь-якому користувачеві за рахунок наявності вказівних коментарів та повідомлення при успішному або не успішному запиті.

До сутності, що відповідає за профіль користувача (таблиця 1) буде заноситься інформація, що в подальшому буде відображатися у даного користувача у профілі, при реєстрації профілю отримує унікальне айді (первинний ключ), так само за допомогою веб-форми реєстрації користувач заносить актуальні особисті дані, що зберігаються під паролем, який кодується у форматі MD5 підвищення безпеки зберігання особистої інформації. За допомогою відповідної форми профіль користувача можна оновити, параметрами, що оновлюються, є всі параметри, крім айді користувача.

Таблиця 1

Параметри сутності «user»

Сутність	Первинний ключ	Атрибути
Користувач (user)	Айді користувача (id);	<ul style="list-style-type: none"> – Ім'я користувача (name); – фамілія користувача (last_name); – email або логін користувача (email); – пароль від профіля користувача шифрований у форматі MD5 (password); – роль користувача (role); – аватар користувача (image).

У сутності, що відповідає за загальну інформацію щодо наявних подій (таблиця 2), заходів тощо інформація заноситься безпосередньо адміністратором

через наявні веб-форми у профілі адміністратора. При створенні запису генерується унікальне айді (первинний ключ), що згодом є незмінним. Доданий адміністратором запис є редагованим лише через базу даних. Веб-форма містить можливість додавати назву, місцезнаходження, дату проведення, ціну та загальний опис. Користувач інформацію з даної таблиці може переглядати безпосередньо в розділі "Події" на сайті, де виведення відбувається в трьох вкладках: "події, що пройшли" (вміщує виведені події минулих 7 днів), "поточні події" (вміщує в собі виведені події за поточні 7 днів) і "майбутні події" (вміщує у собі виведені події за наступні 7 днів).

Таблиця 2

Параметри сутності «ticket»

Сутність	Первинний ключ	Атрибути
Подія або квитка (ticket)	Айді події або квитка (id);	<ul style="list-style-type: none"> – Назва події (nomination); – місце проведення події (location); – дата проведення події (date); – вартість квитка на подію (price); – кратка інформація чи опис події (details).

Якщо користувач був зареєстрований на сайті та авторизований до профілю, то під час перегляду розділу з подіями, внизу буде форма додавання квитка по айді події. Після додавання, у профілі користувача з'являється запис щодо квитка, та до відповідної сутності (таблиця 3) заноситься замовлення зі статусом "Бронювання", далі користувач може скасувати замовлення його видаленням через кнопку видалення у записі, або оплатити його через відповідну форму придбання, що викликається за допомогою кнопки «придбати» під записом з інформацією щодо замовлення. У цій формі є поля введення особистих даних, таких як: ім'я та фамілія замовника (за замовченням беруться дані

авторизованого користувача, однак вони можуть бути змінені), номер телефону, електронна пошта користувача (за замовченням обирається пошта авторизованого користувача, може бути змінена) та дані банківської карти користувача (останній вид даних не заноситься до бази даних за умови безпеки особистої інформації користувачів).

Таблиця 3

Параметри сутності «ticket_order»

Сутність	Первинний ключ	Атрибути
Форма заказів (ticket_order)	Айді заказу (id)	<ul style="list-style-type: none"> – Дата оформлення заказу (time_order); – статус оформлення (status); – індекс айді користувача (user_id); – індекс айді події або квитка (ticket_id); – номер телефону користувача (phonenum); – ім'я на яке оформлюється замовлення (namepersone); – email на який оформлюється заказ (email).

Для можливості подальшого масштабування даної інформаційної системи для розділу новин була зроблена окрема сутність (таблиця 4) для поділу новин на категорії, кожна категорія має унікальний айді і короткий опис. Наявні категорії чотирьох типів: наукові статті, авторські статті, новини обсерваторії та навчальний контент. Реалізовано було додавання статей за категорією авторських статей, інші категорії обираються безпосередньо через базу даних. Тип категорії як інформація для користувачів не виводиться, її можна побачити або перевірити тільки через базу даних.

Параметри сутності «category»

Сутність	Первинний ключ	Атрибути
Категорія (category)	Айді категорії (id)	<ul style="list-style-type: none"> – Назва категорії (categ_name); – загальний опис категорії (categ_description).

В сутності, що відповідає за розділ новин, статей (таблиця 5), інформація заноситься адміністратором сайту через відповідну форму в його профілі на сайті. Після додавання статті вона виводиться у розділі новин, де її та усі раніше публіковані статті може переглянути користувач. Виведення даних йде скорочене: назва статті, зображення до статті, дата публікації та автор. Редагування статей, так само як і з подіями, доступне лише через базу даних.

Параметри сутності «news»

Сутність	Первинний ключ	Атрибути
Новини або статті (news)	Айді новини або статті (news_id)	<ul style="list-style-type: none"> – Дата публікації статті (time_order); – зображення до статті (img); – заголовок статті (title); – повний текст статті (article); – автор публікованої статті (author_name); – індекс айді категорії (category_id).

При переході на конкретну статтю, виводиться вся потокова інформація про неї, що є в таблиці. Під статтею є форма додавання коментарів, з можливістю змінення ім'я користувача, які заносяться в сутність для коментарів користувачів (таблиця 6), і після занесення відображаються під формою для

додавання коментарів. Опубліковані коментарі видаляються лише через базу даних.

Таблиця 6

Параметри сутності «news»

Сутність	Первинний ключ	Атрибути
Коментарі (comment)	Айді коментаря (id)	<ul style="list-style-type: none"> – Текст коментаря (content); – дата публікації коментаря (data_comm); – індекс айді користувача (user_id); – повний текст статті (article); – індекс статті (news_id); – ім'я користувача (user_name).

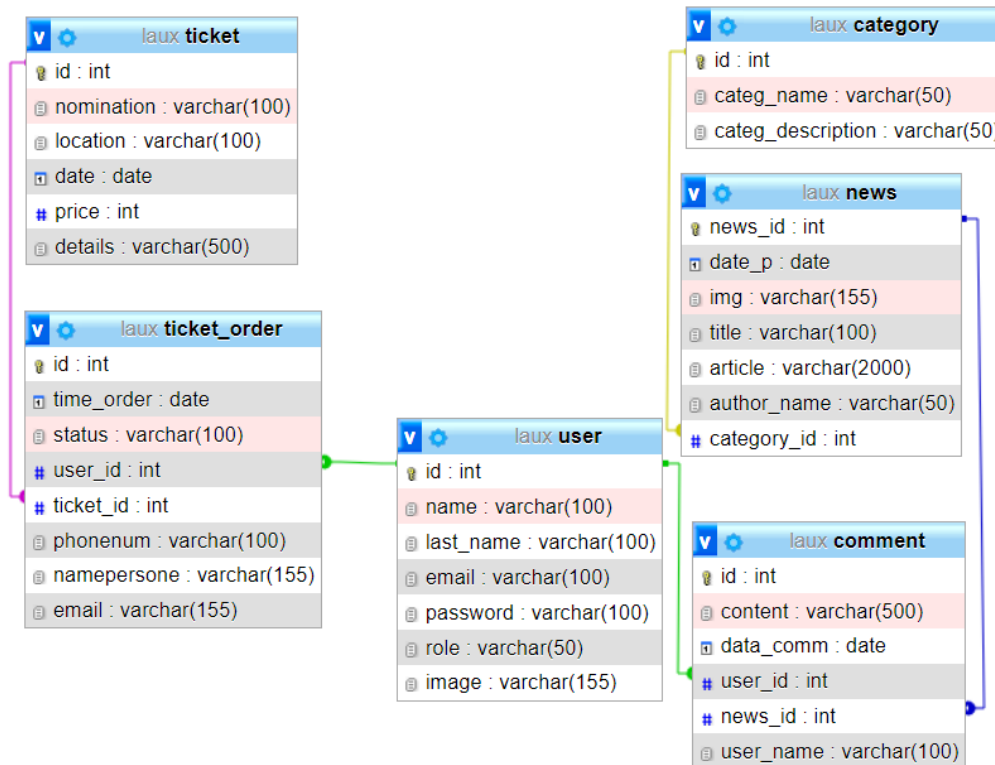


Рис. 2.8. Структура розробленої бази даних

2.5. Обґрунтування та організація вхідних та вихідних даних програми

У розроблюваному веб-орієнтованому застосунку усі вхідні та вихідні дані записуються, зберігаються та виводяться з реляційної бази даних у форматі utf-8, пароль користувача кодується за форматом MD5. Вихідні дані з бази даних оброблюються програмою та виводяться у певному вигляді, щоб користувач мав нагоду взаємодіяти з наявною інформацією. Надалі при заповненні веб-форм користувачем до бази даних надходять вхідні дані.

Наявні види вхідних даних:

- інформація про події, заходи тощо, що заноситься адміністратором;
- дані статей, що заносяться адміністратором;
- інформація у профілі користувача;
- реєстр зареєстрованих користувачів;
- дані, що вводяться користувачем у веб-форми;
- поточна дата користувача;
- оформлені заклази.

Наявні види вихідних даних:

- контент сторінок з розділами та підрозділами;
- інформація про події, заходи для перегляду користувачем;
- прев'ю статті та її зміст для перегляду користувачем;
- заброньовані чи сплачені користувачем квитки на заходи;
- час сходу та заходу Сонця, поточний час користувача.

2.6. Опис розробленої системи

2.6.1. Використані технічні засоби

Для розробки та тестування даного веб-орієнтованого додатку було використано ноутбук, що має такі технічні засоби, як:

- процесор AMD Ryzen 7 3750H 2.30 GHz;

- оперативна пам'ять у розмірі 16 Гб;
- інтегрований графічний процесор AMD Radeon RX Vega 10 Graphics;
- дискретний графічний процесор NVIDIA GeForce RTX 2060 Mobile;
- доступ до ресурсу Інтернет;
- екран з діагоналлю 15,6";
- маніпулятор миша та клавіатура.

Зазначені характеристики не є мінімальними чи рекомендованими.

2.6.2. Використані програмні засоби

Під час розробки та тестування даного веб-орієнтованого додатку було використано такі програмні засоби, як:

- Figma;
- Visual Studio Code;
- Koala;
- Open Server;
- phpMyAdmin.

Figma – онлайн-сервіс, що використовується при розробці інтерфейсу та його елементів за допомогою векторної графіки, та надає можливість не тільки робити статичний дизайн [8], але ще й має функції програмні компоненти, що дозволять реалізувати анімацію та навігацію (імітацію переходу) між розробленими макетами. Інтерфейс програми можна побачити на рис. 2.9.

Головною перевагою є те, що в цьому онлайн-сервісі можна працювати прямо в браузері без необхідності завантажувати додаткове програмне забезпечення на пристрій. Також перевагою є те, що навіть якщо працювати в безкоштовній версії даного програмного забезпечення, функціонал є повним і його вистачить для розробки макетів будь-якого стилю. Створені файли будуть зберігатися в хмарному сховищі самого сервісу, при необхідності будь-який макет, або його елемент можна імпортувати в різних форматах та розмірах.

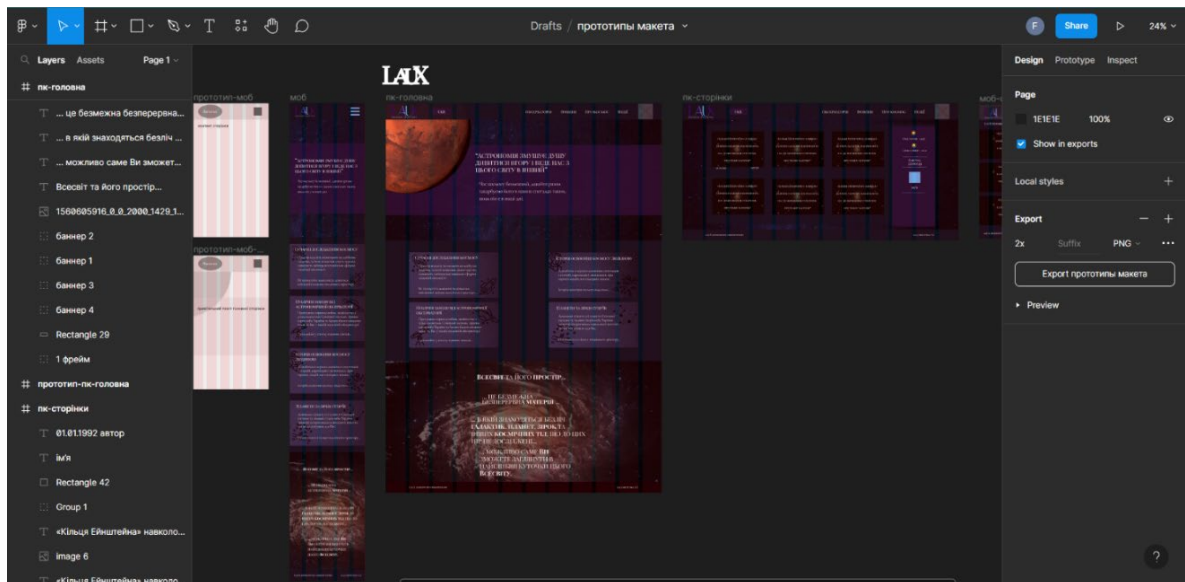


Рис. 2.1. Інтерфейс Figma

Visual Studio Code - є редактором коду, розробленою корпорацією Майкрософт [9]. Підтримує написання коду великою кількістю різноманітних мов програмування, зручний у налаштуванні інтерфейсу, має безліч додаткових плагінів, що розширюють стандартний набір функціоналу. Приклад інтерфейсу можна побачити на рис. 2.10.

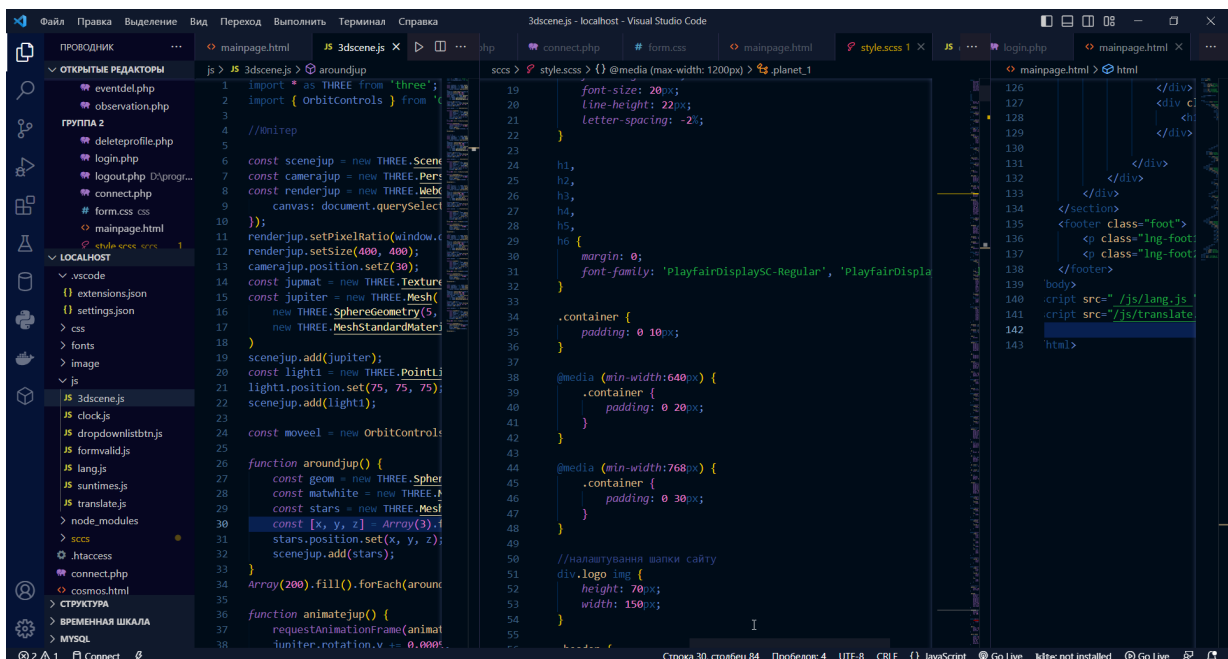


Рис. 2.2. Інтерфейс Visual Studio Code

Koala - це компілятор з відкритим кодом для мов попередньої обробки, таких як: Less, Sass, Compass тощо [10]. Головною перевагою є те, що дана GUI-програма є повністю безкоштовною, зручною в користуванні за рахунок простого зрозумілого інтерфейсу (рис. 2.11), прослуховування файлів, що компілюються, і наступної автоматичної компіляції при будь-якій зміні файлу, так само має функцію автоматичного стиснення коду, що економить місце і робить файли з кодом легкими на сприйняття.

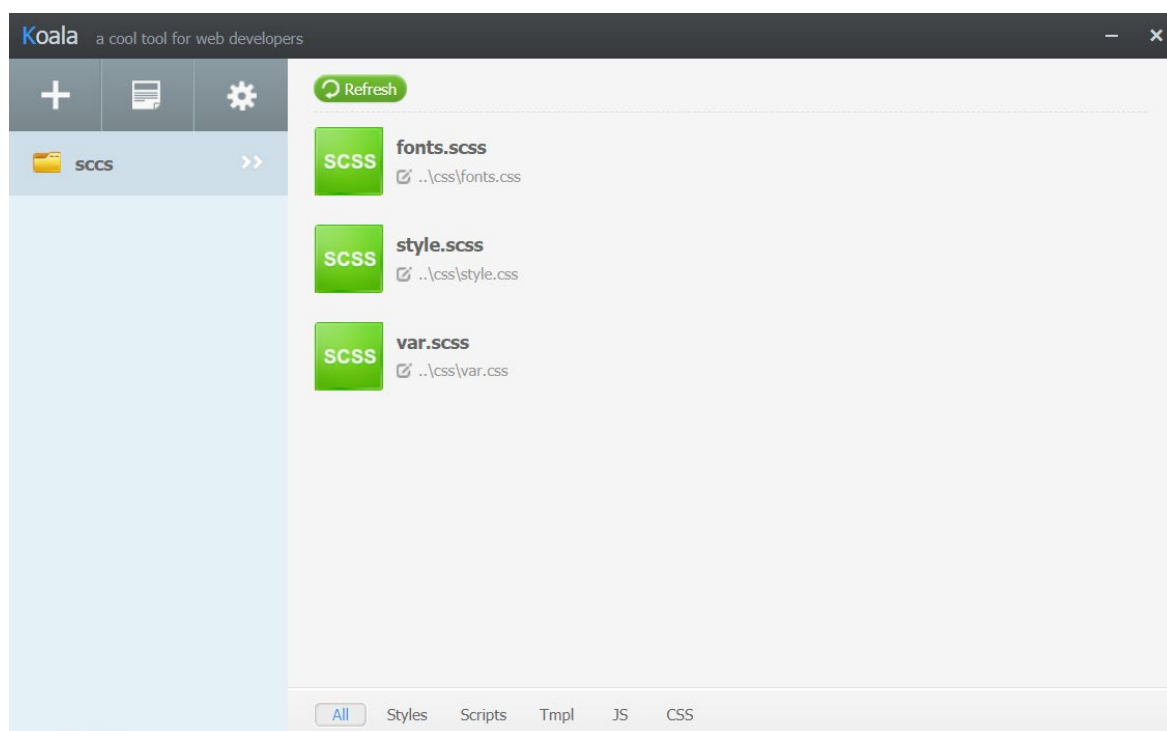


Рис. 2.3. Інтерфейс Koala

Open Server Panel - це портативний програмний комплекс, що містить багато додаткових модулів, наприклад, phpMyAdmin, PHP Adminer та інші [11]. Є безкоштовним програмним забезпеченням, яке дозволяє працювати з мовою PHP. Відображення кореневої папки у запущеному сервері за доменом localhost можна побачити на рис. 2.12. Має низку переваг:

– через те, що програмне забезпечення є портативним, воно не вимагає встановлення до пристрою, що дозволяє його запускати з флешки або зовнішніх жорстких дисків;

- має швидкий відклик на будь-які дії користувача, що пов'язані із зупинкою та завантаженням сервера, так і із застосуванням налаштувань;
- дозволяє змінювати ім'я домену зі стандартного localhost на будь-який присутній у кореневій папці domains.



Рис. 2.4. localhost

phpMyAdmin - це безкоштовний веб-додаток з відкритим вихідним кодом, який використовується для адміністрування баз даних, таких як MySQL та MariaDB [12]. Головною перевагою є робота через веб-інтерфейс, що дає змогу отримувати доступ до баз даних з будь-якого пристрою, який підтримує вихід в інтернет. Використовується для стандартних запитів на мові SQL, таких як: створення бази даних, виконання пошукових запитів, оновлення, додавання облікових записів тощо. Також дозволяє імпортувати дані з файлів різних

форматів і так само експортувати існуючі дані у файли різного формату. Приклад інтерфейсу програми phpMyAdmin наведено на рис. 2.13. Також цей веб-додаток доступний у веб-сервері Open Server Panel, що робить його ще більш доступним і легко впроваджується в проекти, що пишуться мовою програмування PHP.

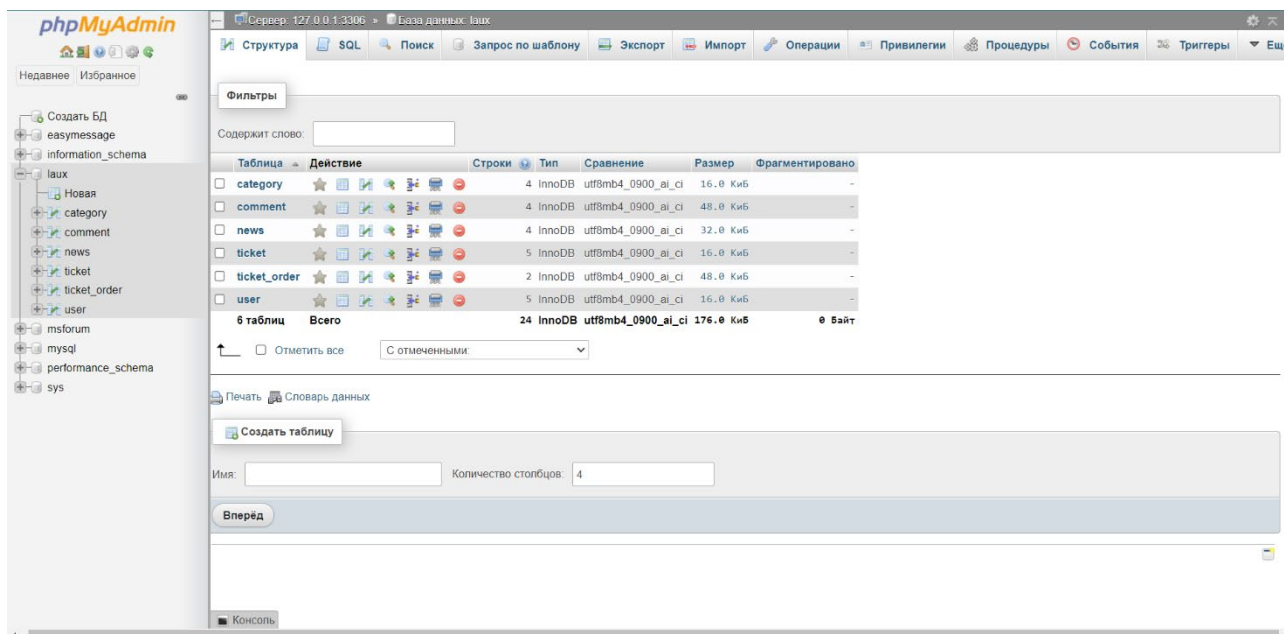


Рис. 2.13. Интерфейс phpMyAdmin

2.6.3. Виклик та завантаження програми

Розробка та тестування веб-орієнтованої програми велася на локальному сервері Open Server Panel, проте для виклику та використання даної інформаційної системи можна використовувати будь-який веб-сервер з підтримкою мови програмування PHP та можливістю його інтерпретувати.

Таким чином можна розглядати узагальнений алгоритм дій для запуску та користування інформаційною системою:

- за наявності серверного обладнання та доменного імені, необхідно встановити на нього php-інтерпретатор;
- зробити базові конфігурації і здійснити підключення до сервера, перемістивши всі потокові файли в кореневу папку сервера;
- перейти по доменному імені.

2.6.4. Опис інтерфейсу користувача

При вході на сайт відобразатиметься головна-лендінг сторінка сайту (рис. 2.14), де відображаються три фрейми: 1) узагальнюючий чи вітальний, 2) що містить у собі банери з активними посиланнями на розділи, 3) інтерактивний контент із можливістю взаємодії.

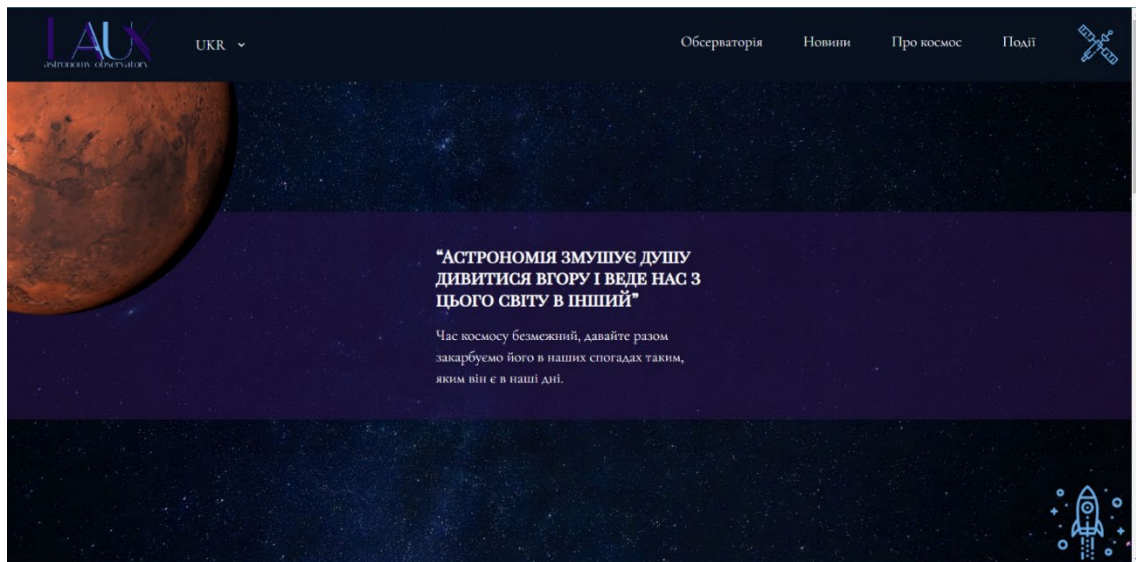


Рис. 2.5. Головна сторінка

Наступний фрейм (рис. 2.15) є ознайомлювальним з наявними розділами на сайті, усі чотири банери мають посилання до відповідних розділів. На самому банері при проведенні курсором з'являється анімація у вигляді неяскравого підсвічування, при наведенні на посилання також працює підсвічування, але вже більш акцентоване.

Наприкінці сторінки є інтерактивний елемент (рис. 2.16 а, б) у вигляді елемента мови HTML - canvas, де малюється сцена з вміщенням 3D-моделей - зірок, що зроблені з множини геометричних тіл типу сфера з вбудованим у бібліотеку THREE.js типом матеріалу MeshBasicMaterial, і в центрі сцени знаходиться геометричне тіло типу сфера з підключеним матеріалом – текстурою Місяця. У цій сцені є автономна плавна анімація зсуву камери за шкалою ординат, так само можна в ручному режимі віддаляти, наближати і крутити саму сцену, не впливаючи при цьому на текст, що знаходиться зверху даного канвасу.



Рис. 2.6. Банери з активними посиланнями

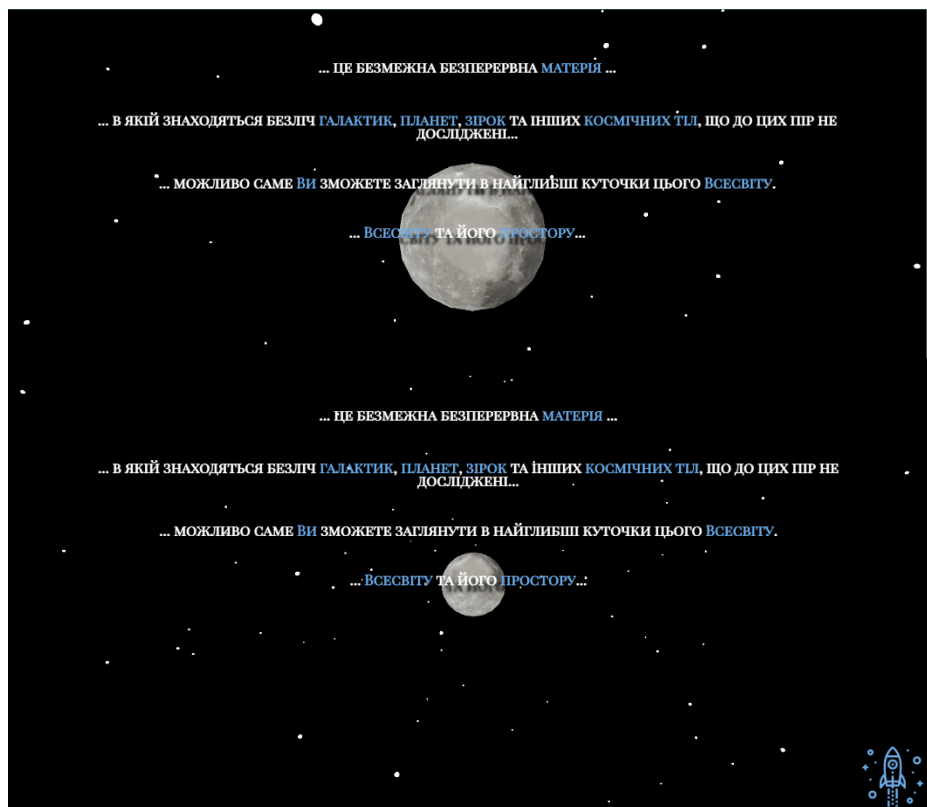


Рис. 2.7 а, б. Приклад взаємодії : а) канвас у момент завантаження, б) канвас після взаємодії з користувачем

Так як було прийнято рішення робити шапку сайту не фіксованою для повнішого відображення інтерактивних елементів, була зроблена кнопка (рис. 2.17), яка дозволяє переміститися вгору сторінки.



Рис. 2.8. Кнопка "до верху"

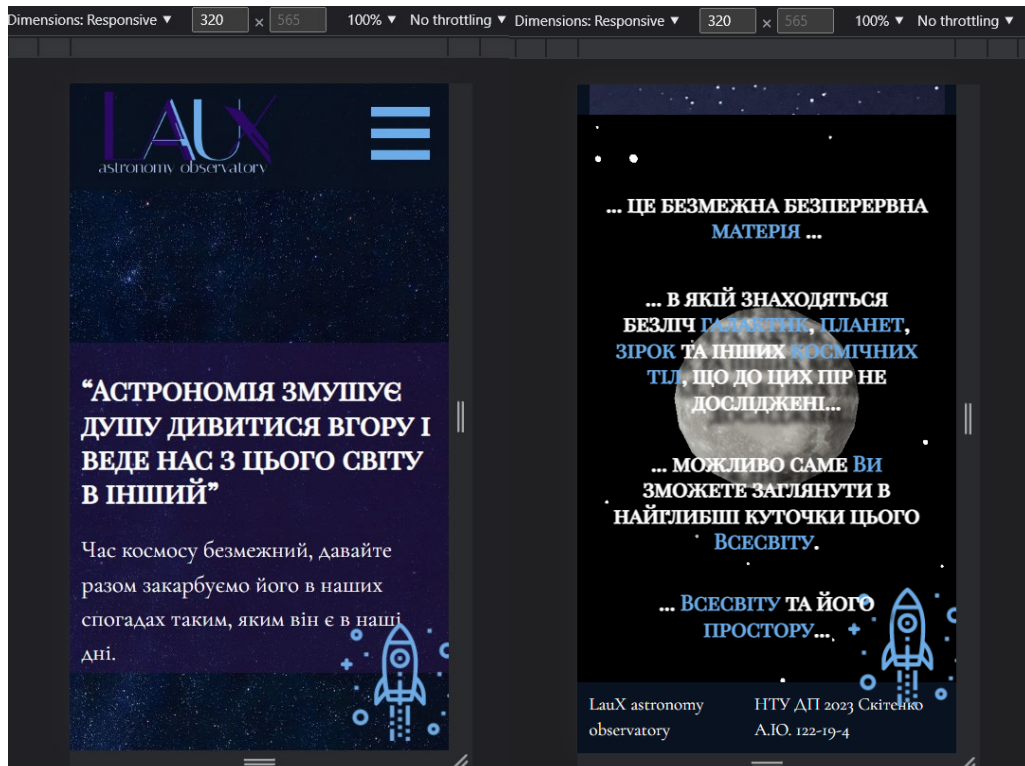


Рис. 2.9 а, б. Приклад адаптивності дизайну: 1) перший фрейм сайту, 2) інтерактивний контент

Основними принципами верстки даного сайту були обрані адаптивність дизайну під різні роздільні здатності екрану (у цьому веб-орієнтованому додатку підтримується роздільність екрана від 320 пікселів шириною до 1920 пікселів шириною) та мультимовність, тобто підтримка англійської та української мов. Головна сторінка є сторінкою з оригінальним дизайном, так як усі внутрішні сторінки мають один відмінний шаблон від цього, приклад дизайну у мобільної

версії наведено на рис. 2.18 а, б інтерактивна частина повністю зберігає свій функціонал.

Вибрати іншу мову можна у шапці сайту за допомогою спеціального селектора змінюється мова на сайті, так само її можна змінити через додавання розкладки з ключем "#" (рис. 2.19 а, б). Приклад перекладу сторінки наведено на рис. 2.20.

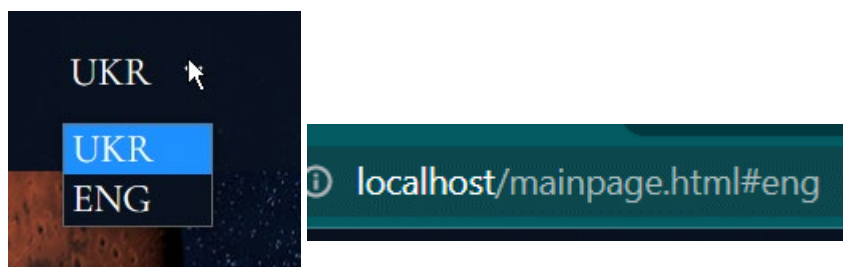


Рис. 2.10 а, б. Мультимовність: а) селектор для змінення мови, б) індикатор обраної мови

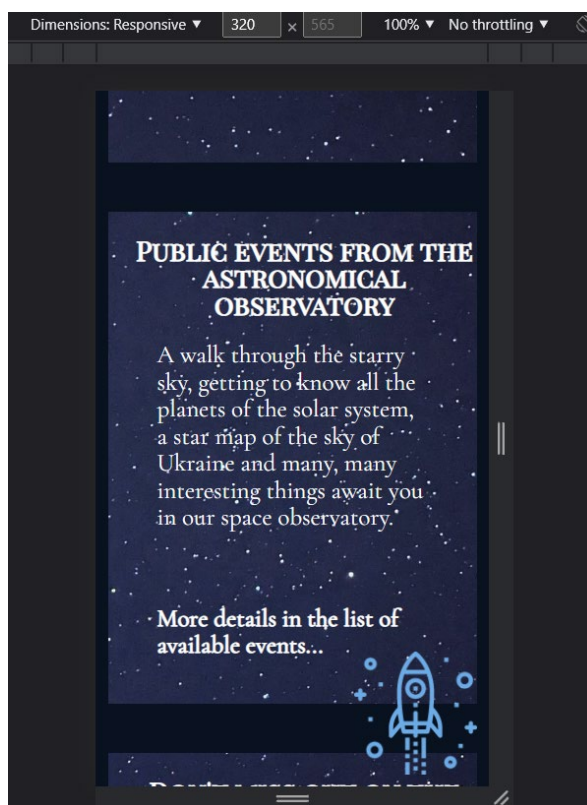


Рис. 2.11. Приклад переведення сторінки на англійську мову у мобільній версії

Далі через шапку сайту (рис. 2.21) можна перейти до інших розділів навігаційного меню. Першим розділом у шапці буде "Обсерваторія" (рис. 2.23), це розділ, де є загальна інформація щодо обсерваторій та відомих вчених-винахідників у сфері астрономії та астрофізики. Перехід на головну сторінку здійснюється через натискання на логотип. У мобільній версії посилання навігації згортаються у випадаючий список, що відкривається по натисканню на кнопку у шапці.



Рис. 2.12. Шапка сайту

Також у сайту присутній футер (рис. 2.22), де є назва сайту та його розробник.



Рис. 2.22. Футер сайту

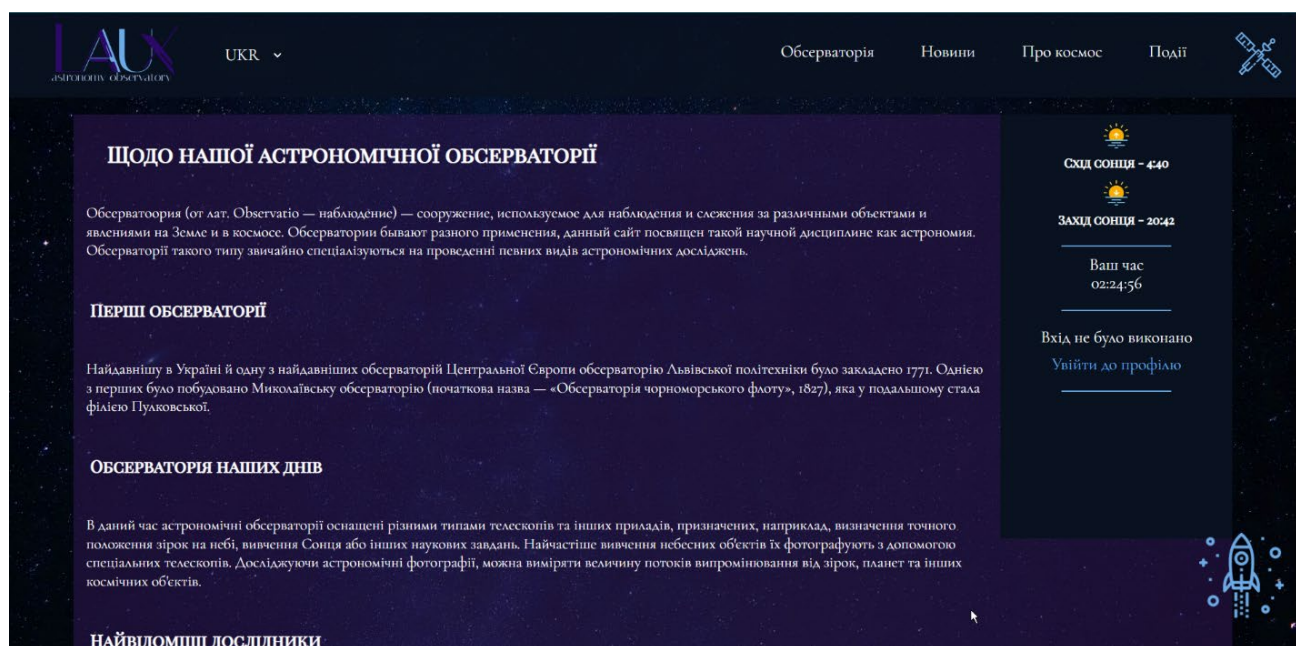


Рис. 2.23. Розділ "Обсерваторія"

На внутрішніх сторінках з'являється меню-бар, який містить у собі: 1) час сходу та заходу сонця, що прораховується за поточною датою та місцезнаходженням (за замовчуванням було взято місто Дніпро), 2) годинник (рис. 2.24), 3) індикатор входу (якщо користувач не був авторизований, то буде посилання на профіль користувача для подальшої реєстрації, якщо користувач був авторизований, то буде виведена мініатюра його аватару та ім'я) (рис. 2.25 а, б).

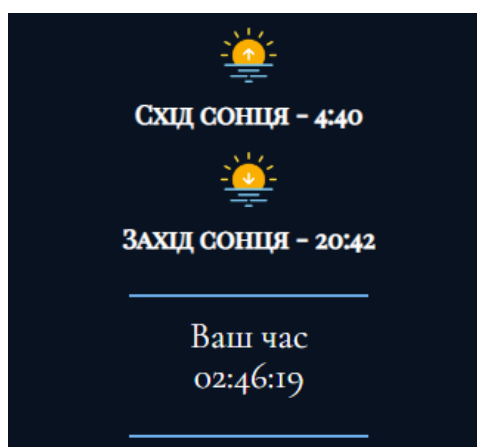


Рис. 2.13. Меню-бар: змінні часу

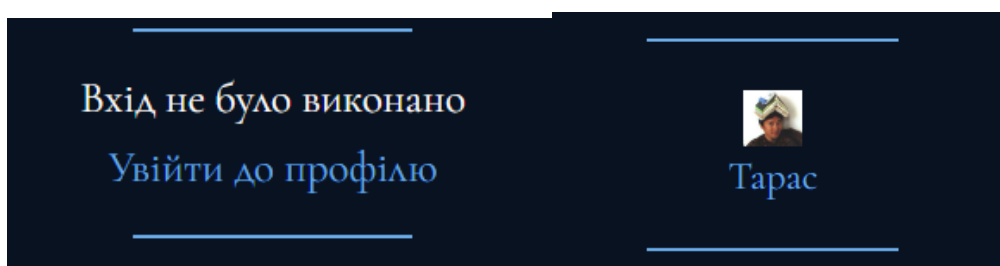


Рис. 2.14 а, б. Індикатор входу: а) при неавторизованому користувачі, б) користувача авторизовано

При переході у мобільну версію сайту (рис. 2.26) меню-бар зникає, та вгорі тіла сайту з'являється навігація "хлібні крихти" з поточним відкритим розділом.

Наступним розділом навігації в шапці є розділ "Новини" (рис. 2.27). Усі записи, які є, виводяться з бази даних, назва статті – це посилання на неї, при натисканні відкривається сторінка з відповідною статтею (рис. 2.28) та полем

коментарів для користувачів як зареєстрованих, як і ні. З коментарями присутні два можливі способи впливу: публікація та видалення (рис. 2.29). Публікувати коментарі може будь який користувач, однак видаляти може лише користувач, що його опублікував. Якщо коментарів під статтею немає, то відображається повідомлення «Коментарів немає!».



Рис. 2.26. Вигляд у мобільній версії

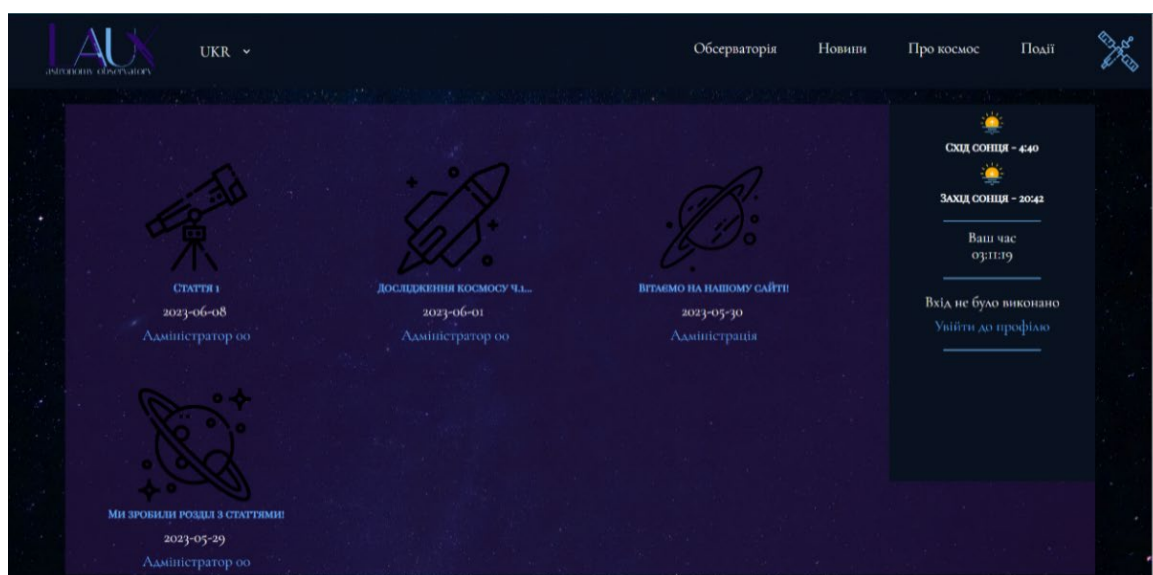


Рис. 2.27. Повний вигляд сторінки

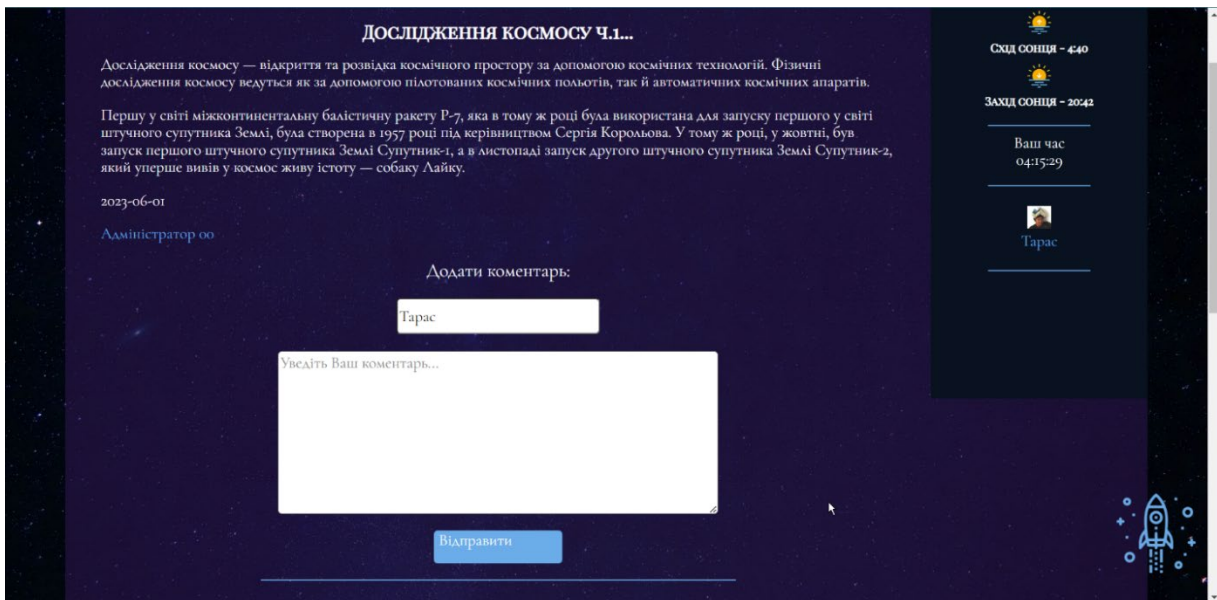


Рис. 2.28. Сторінка статті

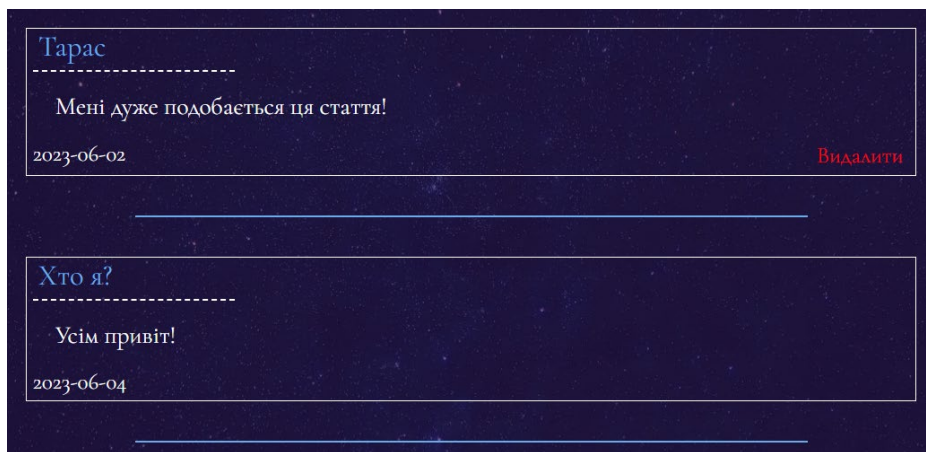


Рис. 2.29. Оpubліковані коментарі

Наступним розділом йде "Про космос", цей розділ містить у собі інтерактивно-навчальний контент з інформацією про планети Сонячної системи, популярні сузір'я зіркової карти та модель руху планет навколо Сонця.

У банері для перегляду короткої інформації розміщено п'ять перших від Сонця планет: Меркурій, Венера, Земля, Марс та Юпітер (рис. 2.30). При наведенні на якусь планету, вона знизу підсвічується блакитним кольором (рис. 2.31), при натисканні на планету відкривається вікно, що спливає (рис. 2.32), із загальною інформацією про планету і канвасом зі сценою, де дана планета

анімована на плавний рух по своїй осі, її так само можна наближати, віддаляти, крутити додатково.



Рис. 2.30. Банер з планетами та сузір'ями



Рис. 2.31. Планета під час наведення

Наступний банер (рис. 2.33 а, б) включає канвас, де реалізована анімація обертання всіх планет Сонячної системи (включаючи Плутон) навколо Сонця, яке так само обертається навколо своєї осі, з частковою передачею фізичних особливостей, таких як: відмінність між швидкостями обертання планет, відстанню між космічними тілами, різницею між розмірами і масами планет.

Крім автономної анімації сцени, також є можливість покрутити, збільшити, або віддалити об'єкти сцени.

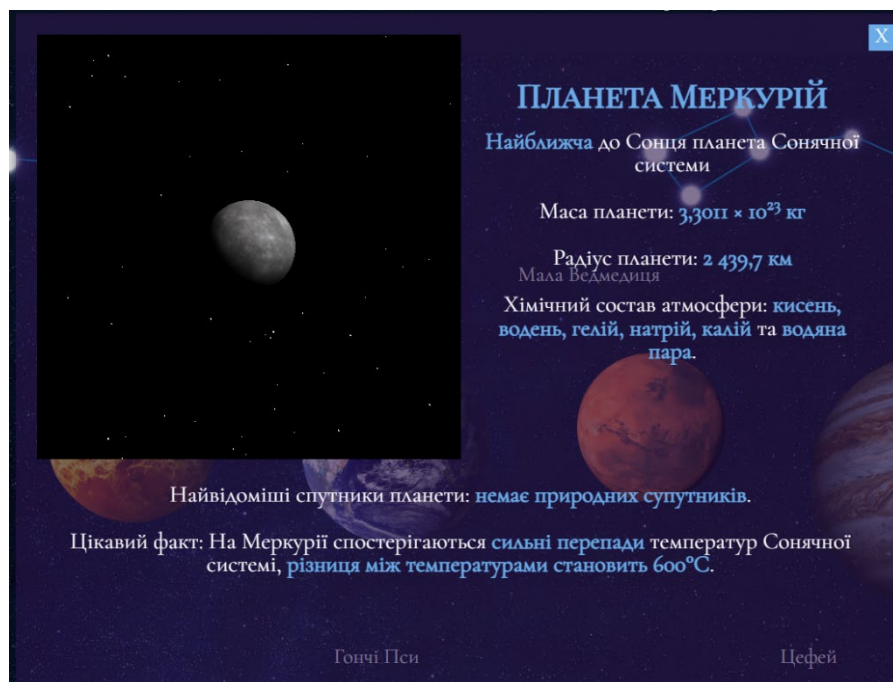


Рис. 2.32. Спливаюче вікно з інформацією

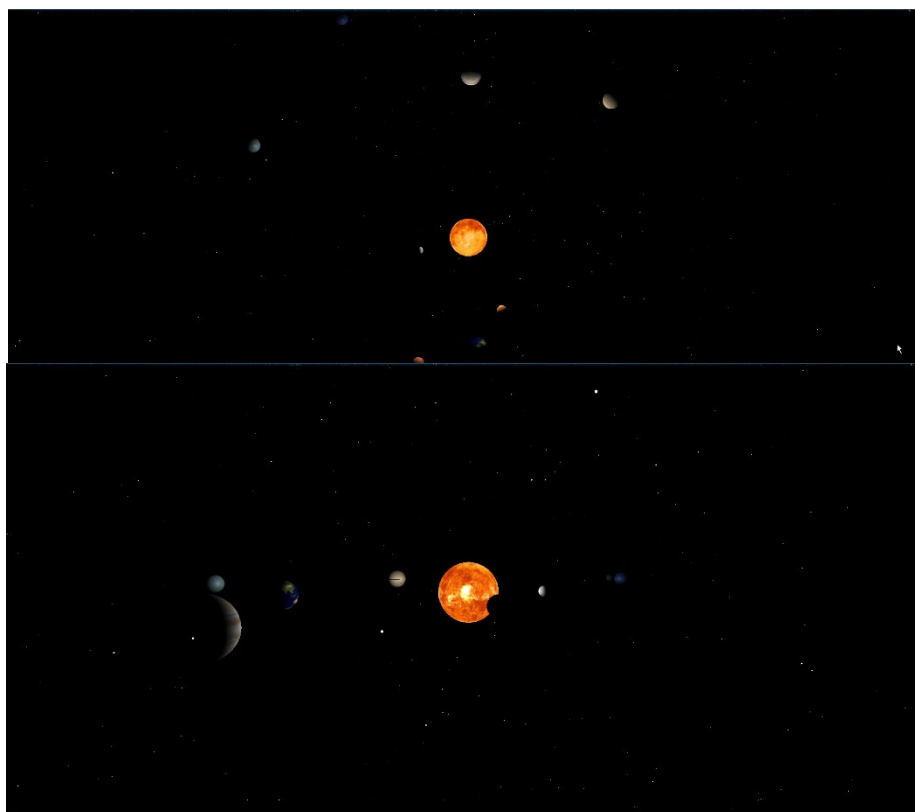


Рис. 2.33 а, б. Анімована модель Сонячної системи: а) початкове положення камери, б) після взаємодії

Передостаннім розділом є розділ "Події" (рис. 2.34), що містить у собі виведену з бази даних інформацію щодо заходів за певну кількість часу:

– "попередні події" - всі події за минулі сім днів, давніші події з бази даних не виводяться;

– "поточні події" - ті заходи, що відбуваються протягом найближчих семи днів починаючи з поточної дати;

– "майбутні події" - ті заходи, що будуть йти протягом тижня після поточного тижня.

Виведення з бази даних було зроблено в стилі списку, що випадає: при натисканні - відображаються всі запити, що задовольняють вимогам. У кожного заходу є порядковий номер, через який авторизовані користувачі можуть додати його собі в профіль для подальшої оплати або видалення замовлення. Якщо користувача не було авторизовано, форма для додавання у нього буде відсутня.

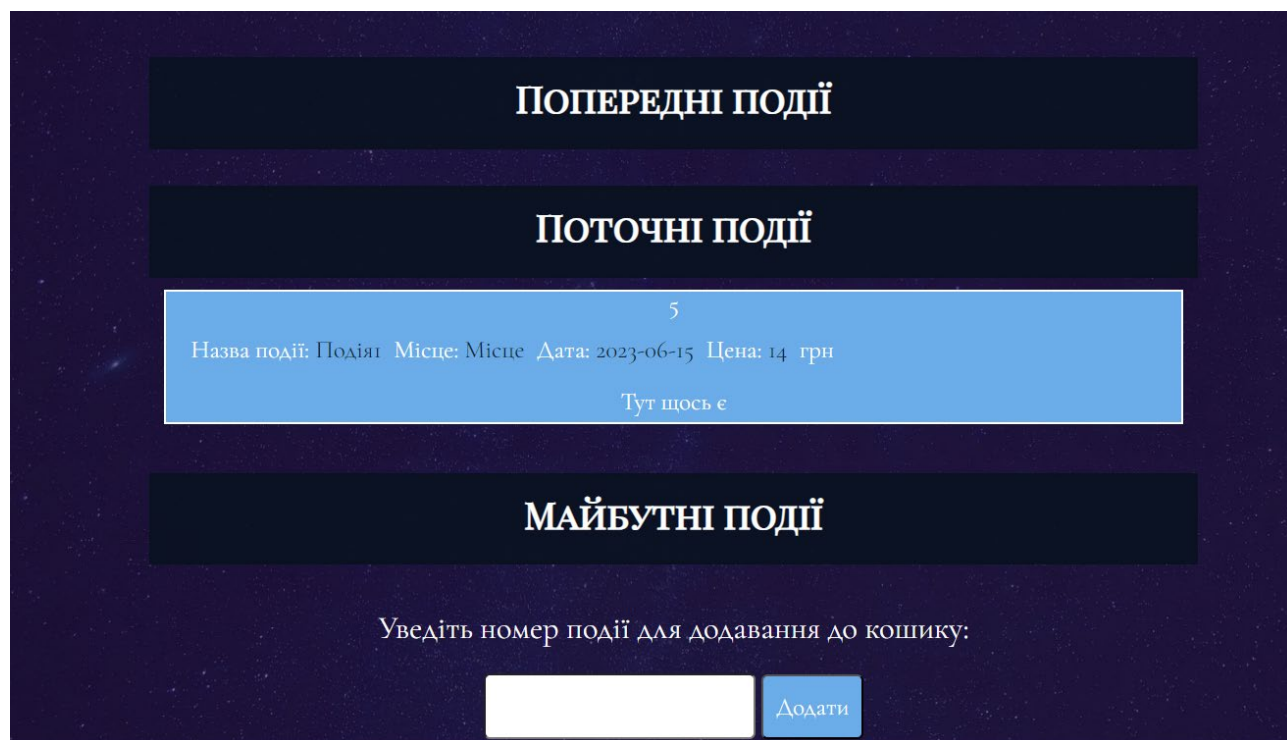


Рис. 2.34. Розділ "події" без меню-бара

Останній розділ - "Профіль", посиланням на нього є іконка в кінці шапки, або якщо з мобільної версії сайту, то однойменне посилання в кінці списку. Для реєстрації (рис. 2.35) необхідно заповнити форму, яка валідується, доки форма

не буде перевірена на правильність, кнопка реєстрації буде заблокована. Такого ж формату валідація (рис. 2.36 а, б) присутня і у формі входу, або авторизації користувача.

Уведіть Ваші ім'я та фамілію:

Уведіть Ваше ім'я

Уведіть Вашу фамілію

Уведіть Ваш email:

Уведіть Ваш email

Уведіть пароль:

Уведіть Ваш пароль

Уведіть пароль повторно:

Підтвердіть Ваш пароль

Додайте Ваш аватар:

Виберите файл

Файл...бран

Перевірити

Реєстрація

Ви вже були зареєстровані? [Увійти](#)

Рис. 2.35. Форма реєстрації

Уведіть Ваш email:

Уведіть Ваш email

Уведіть пароль:

Уведіть Ваш пароль

Перевірити

Увійти

Ви ще не зареєстровані? [Зареєструватися](#)

Уведіть Ваш email:

Уведіть Ваш email

Будь ласка уведіть дані!

Уведіть пароль:

Уведіть Ваш пароль

Мінімальна кількість символів: 5

Перевірити

Увійти

Рис. 2.17 а, б. Форма входу: а) у початковому стані, б) при проваленій валідації

Вже авторизований користувач (вигляд профіля можна побачити на рис. 2.37) може впливати зі своїм профілем: редагувати, видаляти та виходити з нього. За умови, що користувач додав у кошик квитки на заходи, у профілі під загальною інформацією з'являються поля замовлення, які можна видалити або придбати через форму. При правильному та повному заповненні форми та успішного запиту до бази даних на замовлення ставиться статус "Оплачено". Форми редагування та придбання можна побачити на рис. 2.38 а, б.

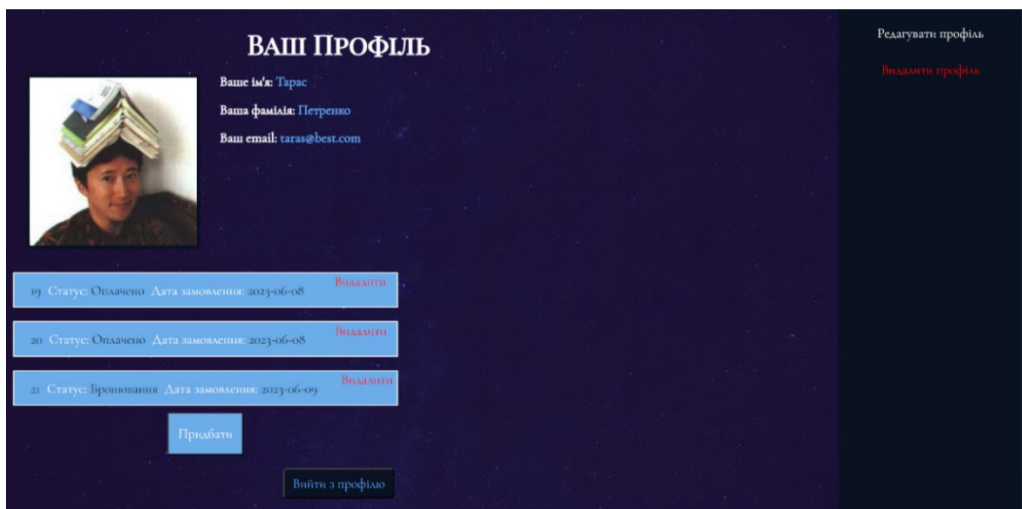


Рис. 2.37. Профіль користувача при авторизації

Рис. 2.38 а, б. Форми: а) оновлення даних профіля, б) придбання квитків

Всі розглянуті вище способи взаємодії з базою даної та веб-формами належали до клієнтської частини, тобто безпосередньо до користувача. Далі йтиметься про функціонал для адміністратора. Функція додавання контенту у вигляді записів у розділі новин та подій на веб-сайт здійснюється через профіль адміністратора (рис. 2.39), а редагування вже існуючого контенту у розділі новин відбувається безпосередньо через сторінку статті (рис. 2.40).

ПАНЕЛЬ ДОДАВАННЯ КОНТЕНТУ

ДОДАВАННЯ ПОДІЙ:

Назва події:

Місце проведення події:

Дата проведення події:

Вартість відвідування:

Додати коментар:

ДОДАВАННЯ СТАТТІ:

Назва статті:

Дата публікації:

Зображення до статті:

Додати коментар:

Рис. 2.39. Додавання контенту через профіль адміністратора

При переході на опубліковані статті буде відкриватися форма для її редагування та подальшого збереження.

Дослідження космосу ч.1...

<р> Дослідження космосу — відкриття та розвідка космічного простору за допомогою космічних технологій. Фізичні дослідження космосу ведуться як за допомогою пілотованих космічних польотів, так й автоматичних космічних апаратів. <р>

<р>Першу у світі міжконтинентальну балістичну ракету Р-7, яка в тому ж році була використана для запуску першого у світі штучного супутника Землі, була створена в 1957 році під керівництвом Сергія Корольова. У тому ж році, у жовтні, був запуск першого штучного супутника Землі Супутник-1, а в листопаді запуск другого штучного супутника Землі Супутник-2, який уперше вивів у космос живу істоту — собаку Лайку. <р>

Рис. 2.40. Форма редагування статті

РОЗДІЛ 3

ЕКОНОМІЧНИЙ РОЗДІЛ

3.1. Розрахунок трудомісткості та вартості розробки інформаційної системи

Вхідні дані для обрахування:

- передбачуване число операторів – 2508 строк;
- коефіцієнт корекції програми в ході її розробки – 0,4;
- коефіцієнт складності програми – 1,3;
- годинна заробітна плата програміста – 108,51 грн/год;
- коефіцієнт збільшення витрат праці в наслідок недостатнього опису задачі – 1,4;
- коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності – 1;
- вартість машино-години ЕОМ – 19,3 грн/год;

Згідно зі статистикою заробітна плата back-end і front-end розробника приблизно однакові, таким чином за основу братиметься мінімальна ставка джуніора front-end розробки, що в Україні на травень місяць 2023 року, за порталом пошуку вакансій “Jooble” становить у середньому близько 17 тис. грн. на місяць, а погодинно – 108,51 грн [17].

Для можливості розробляти інформаційну систему потрібен досить потужний комп'ютер із постійним підключенням до Інтернету та електроенергії. Тариф за електроенергію для побутових споживачів – 2,64 грн. (з ПДВ) за 1 кВт·годину [18], оренда ноутбука на 62 дні коштуватиме 3420 грн [19]. Пакет домашнього інтернету коштує 150 грн на місяць. У середньому ноутбук споживає 40 Ватт на годину, загальноприйнята кількість робочих годин на тиждень – 159 годин, таким чином за електроенергію на годину йтиме 16,79 грн. З урахуванням усіх показників собівартість машино-години роботи ЕОМ складатиме 19,3 грн.

Нормування праці в процесі створення ПЗ істотно ускладнено в силу творчого характеру праці програміста. Тому трудомісткість розробки ПЗ може бути розрахована на основі системи моделей з різною точністю оцінки.

Трудомісткість розробки ПЗ можна розрахувати за формулою:

$$t = t_o + t_u + t_a + t_n + t_{oml} + t_d, \text{ людино-годин,} \quad (3.1)$$

де t_o - витрати праці на підготовку й опис поставленої задачі (приймається 50 людино-годин);

t_u - витрати праці на дослідження алгоритму рішення задачі;

t_a - витрати праці на розробку блок-схеми алгоритму;

t_n -витрати праці на програмування по готовій блок-схемі;

t_{oml} -витрати праці на налагодження програми на ЕОМ;

t_d - витрати праці на підготовку документації.

Складові витрати праці визначаються через умовне число операторів у програмному забезпеченні, що розробляється.

Умовне число операторів:

$$Q = q \cdot C \cdot (1 + p), \quad (3.2)$$

де q - передбачуване число операторів (за умовами - 2508);

C - коефіцієнт складності програми (за умовами - 1,3);

p - коефіцієнт корекції програми в ході її розробки (за умовами - 0,4).

Розрахуємо умовне число операторів або підпрограм наявної інформаційної системи за формулою (3.2):

$$Q = 2508 * 1,3 * (1 + 0,4) = 4564,56.$$

Витрати праці на вивчення опису задачі t_u визначається з урахуванням уточнення опису і кваліфікації програміста:

$$t_u = \frac{Q * B}{(75..85) * k}, \quad (3.3)$$

де В - коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі.

к - коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності. При стажі роботи часом від 1 до 2 років коефіцієнт дорівнює 1,1.

Збільшення витрат праці внаслідок недостатнього опису завдання приймається не більше 50%, значення В було прийнято як 1,4.

З урахуванням коефіцієнта кваліфікації $k = 1,1$, за формулою (3.3) отримуємо витрати праці на вивчення опису завдання:

$$t_u = \frac{4564,56 * 1,4}{85 * 1,1} = 68,35, \text{ людино-годин.}$$

Витрати праці на розробку алгоритму рішення задачі визначаються за формулою:

$$t_a = \frac{Q}{(20...25) * k}, \quad (3.4)$$

де Q – умовне число операторів інформаційної системи (за умовами - 2508);

к – коефіцієнт кваліфікації програміста (за умовами - 1,1).

Розрахуємо витрати праці на розробку алгоритму рішення задачі за формулою (3.4):

$$t_a = \frac{4564,56}{25 * 1,1} = 165,98, \text{ людино-годин.}$$

Витрати на складання програми по готовій блок-схемі:

$$t_n = \frac{Q}{(20...25)*k}, \quad (3.5)$$

де Q – умовне число операторів інформаційної системи (за умовами - 2508);

Здійснимо розрахунки за формулою (3.5):

$$t_n = \frac{4564,56}{25*1,1} = 165,98, \text{ людино-годин.}$$

Витрати праці на налагодження програми на ЕОМ:

– за умови автономного налагодження одного завдання:

$$t_{\text{отл}} = \frac{Q}{(4..5)*k}, \quad (3.6)$$

– за умови комплексного налагодження завдання:

$$t_{\text{отл}}^k = 1,5 * t_{\text{отл}}, \quad (3.7)$$

За формулами (3.6) та (3.7) розрахуємо показники витрати праці на налагодження програми на ЕОМ за умови автономного налагодження одного завдання та умови комплексного налагодження завдання відповідно.

$$t_{\text{отл}} = \frac{4564,56}{5*1,1} = 829,92, \text{ людино-годин,}$$

$$t_{\text{отл}}^k = 1,5 * 829,92 = 1244,88, \text{ людино-годин.}$$

Витрати праці на підготовку документації визначаються за формулою:

$$t_d = t_{\text{др}} + t_{\text{до}}, \quad (3.8)$$

де $t_{\text{др}}$ - трудомісткість підготовки матеріалів і рукопису:

$$t_{др} = \frac{q}{(15..20)*k}, \quad (3.9)$$

$t_{до}$ - трудомісткість редагування, печатки й оформлення документації:

$$t_{до} = 0,75 * t_{др}, \quad (3.10)$$

Проведемо розрахунки за формулами (3.8)-(3.10) для визначення витрат праці на підготовку документації:

$$t_{др} = \frac{4564,56}{20*1,1} = 207,48, \text{ людино-годин,}$$

$$t_{до} = 0,75 * 207,48 = 155,61, \text{ людино-годин,}$$

$$t_{д} = 207,48 + 155,61 = 363,09, \text{ людино-годин.}$$

Маючи усі розрахунки скористаємося формулою (3.1) для отримання повної оцінки трудомісткості розробки інформаційної системи:

$$t = 50 + 68,35 + 165,98 + 165,98 + 829,92 + 363,09 =$$

$$= 1643,32, \text{ людино-годин.}$$

3.2. Розрахунок витрат на створення інформаційної системи

Витрати на створення ПЗ КПО включають витрати на заробітну плату виконавця програми $Z_{зп}$ і витрат машинного часу, необхідного на налагодження програми на ЕОМ:

$$K_{по} = Z_{зп} + Z_{мв}, \text{ грн.} \quad (3.11)$$

Заробітна плата робітників визначається за формулою:

$$З_{зп} = t * C_{пр}, \text{ грн}, \quad (3.12)$$

де t - загальна трудомісткість, людино-годин;

$C_{пр}$ - середня годинна заробітна плата програміста, грн/година.

Вартість машинного часу, необхідного для налагодження програми на ЕОМ, визначається за формулою:

$$З_{мв} = t_{отл} * C_{мч}, \text{ грн}, \quad (3.13)$$

де $t_{отл}$ - трудомісткість налагодження програми на ЕОМ, год (за розрахунками 829,92 людино-годин);

$C_{мч}$ - вартість машино-години ЕОМ, грн/год (за умовами 19,3 грн/год).

За умовами того, що t дорівнює 1643,32 людино-годин та погодинної заробітної плати 108,51 грн/год, розрахуємо загальну заробітну плату за формулою (3.12):

$$З_{зп} = 1643,32 * 108,51 = 178316,65, \text{ грн.}$$

За формулою (3.13) розрахуємо вартість машинного часу, необхідного для налагодження програми на ЕОМ:

$$З_{мв} = 829,92 * 19,3 = 16017,46, \text{ грн.}$$

Підставимо усі розрахунки у формулу (3.11) для розрахування витрат на створення ПЗ КПО:

$$К_{пю} = 178316,65 + 16017,46 = 194334,11, \text{ грн.}$$

Очікуваний період створення програмного застосунку:

$$T = \frac{t}{B_k * F_p}, \text{ міс,} \quad (3.14)$$

де B_k - число виконавців;

F_p - місячний фонд робочого часу (при 40 годинному робочому тижні $F_p = 176$ годин).

Розрахуємо очікуваний період створення програмного застосунку за формулою (3.14):

$$T = \frac{1643,32}{1 * 176} \approx 9,3 \text{ міс.}$$

Висновки: за проведеними розрахунками вище, вартість розробки веб-орієнтованого застосунку для астрономічної обсерваторії становить 194334,11 грн, та не вимагає додаткових трат. Трудомісткість розробки дорівнює 1643,32 людино-години. Також по розрахунках очікуваний період розробки інформаційної системи дорівнює 9,3 місяці, цей результат залежить від кількості строк або числа операторів, що було написано у ході розробки програмного застосунку, коефіцієнту кваліфікації розробника, та також включає у себе такі параметри, як: час на дослідження, час на розробку вирішення поставленого завдання, час витрачений на програмування за готовим алгоритмом тощо.

ВИСНОВКИ

Метою кваліфікаційної роботи було розробка веб-орієнтованого застосунку для астрономічної обсерваторії засобами мов HTML, CSS, JavaScript та PHP.

Актуальність теми пов'язана зі стрімким розвитком космічних технологій та їх поширення, зростає потреба в інформації у сфері науки астрономія. З кожним роком ця сфера набуває все більшої популярності, а космічні технології стають серйозним фактором, що впливає на різні структури та тенденції. Саме це обумовлює загальний інтерес громадськості у подальшому розвитку цієї сфери.

Практичне значення полягає у наданні інтерактивно-навчального контенту користувачам сайту будь-якої вікової категорії, також дана розробка надає доступ до порталу новин, що є на веб-сайті, і дає можливість дистанційно переглядати всі наявні заходи, події тощо, бронювати чи купувати їх. Під час виконання даної роботи були виконані такі поставлені задачі, як:

- аналіз предметної області задачі та наявних технічних засобів для її вирішення;
- впровадження усіх функцій, що було передбачено постановкою завдання;
- розробка веб-орієнтованого застосунку;
- загальний опис застосованих методів, технологій тощо.

Повний перелік доступних функцій розробленого застосунку:

- адаптація інтерфейсу користувача відносно роздільної здатності екрану;
- змінення локалізації (підтримка української та англійської мов);
- перегляд наявних заходів, подій тощо, бронювання або купівля квитків до них;
- перегляд такого інтерактивно-навчального контенту на тему космічного простору, як: кратка інформація щодо планет Сонячної системи, можливість взаємодії з наявними 3D-моделями цих планет, макет обертання планет навколо Сонця та перегляд популярних сузір'їв зоряної карти;
- перегляд та коментування у розділі новин на застосунку;

– додавання та редагування статей до розділу новин адміністратором, також додавання нових записів щодо подій, заходів.

Також у кваліфікаційній роботі було розраховано економічну частину, а саме: вартість розробки (за розрахунками складає - 194334,11 грн), трудоміскість (за розрахунками складає - 1643,32 людино-години) та по розраховано очікуваний період розробки, що дорівнює 9,3 місяці.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Глумачний словник з інформатики / Г.Г. Півняк, Б.С. Бусигін, М.М. Дівізінюк та ін. – Д., Нац. гірнич. ун-т, 2010. – 600 с.
2. Shelley P. JavaScript Cookbook: Programming the Web, Second Edition / USA, California, 2015. – 634 p.
3. WHATWG community – HTML Living Standard - 4.3 Sections. URL: <https://html.spec.whatwg.org/multipage/sections.html> (дата звернення: 05.05.2023).
4. What is Client/Server Architecture? - Definition from Techopedia. URL: <https://www.techopedia.com/definition/438/clientserver-architecture> (дата звернення: 12.05.2023).
5. Goodman D. Dynamic HTML: The Definitive Reference: A Comprehensive Resource for XHTML, CSS, DOM, JavaScript, Third Edition / USA, California, 2007. – 1322 p.
6. Sass: Documentation. <https://sass-lang.com/documentation/> (дата звернення: 18.05.2023).
7. Fundamentals - three.js manual. <https://threejs.org/manual/#en/fundamentals> (дата звернення: 20.05.2023).
8. Developer onboarding Guide for Figma. <https://www.figma.com/best-practices/guide-to-developer-handoff/components-styles-and-documentation/> (дата звернення: 22.05.2023).
9. Visual Studio Code – Code Editor | Microsoft Azure. <https://azure.microsoft.com/en-gb/products/visual-studio-code> (дата звернення: 22.05.2023).
10. Docs oklai/koala Wiki GitHub. Getting Started with Koala. <https://github.com/oklai/koala/wiki/Docs#getting-started-with-koala> (дата звернення: 22.05.2023).
11. Документація / Open Server Panel. <https://ospanel.io/docs/> (дата звернення: 23.05.2023).

12. Introduction — phpMyAdmin 5.1.4 documentation. <https://docs.phpmyadmin.net/en/latest/intro.html> (дата звернення: 23.05.2023).
13. Robin Nixon. Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5 / USA, California, 2018. – 830 p.
14. Jon Duckett. PHP & MySQL: Server-side Web Development 1st Edition / USA, New Jersey, 2022. – 672 p.
15. Adobe Photoshop | Common Questions. FAQ. <https://helpx.adobe.com/photoshop/faq.html> (дата звернення: 26.05.2023).
16. Ikram Hawramani. HTML, CSS & JavaScript for Complete Beginners: A Step by Step Guide to Learning HTML5, CSS3 and the JavaScript Programming Language / USA, Arizona, 2018. – 581 p.
17. Junior front-end developer зарплата - Перевірити середню junior front-end developer ставку на Jooble. <https://ua.jooble.org/salary/junior-front+end-developer#hourly> (дата звернення: 04.06.2023).
18. YASNO - Тарифи для населення. <https://yasno.com.ua/b2c-tariffs> (дата звернення: 04.06.2023).
19. Оренда ноутбуків у місті Дніпро. <https://shop.vortex.dp.ua/ua/cp65737-arenda-noutbukov.html> (дата звернення: 04.06.2023).
20. Creating a scene — three.js docs. <https://threejs.org/docs/#manual/en/introduction/Creating-a-scene> (дата звернення: 07.06.2023).
21. Jos Dirksen. Learning Three.js: The JavaScript 3D Library for WebGL / UK, Birmingham, 2013. – 402 p.

ЛІСТИНГ ПРОГРАМИ

```

//Головна сторінка
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <link rel="stylesheet" href="css/reset.css">
  <link rel="stylesheet" href="css/style.css">
  <script src="js/dropdownlistbtn.js"></script>
  <script type="importmap">
    { "imports": { "three": "https://unpkg.com/three@0.138.0/build/three.module.js", "OrbitControls":
"https://unpkg.com/three@0.138.0/examples/jsm/controls/OrbitControls.js" } }
  </script>

  <title>LauX</title>
</head>

<body>
  <a href="#top"></a>
  <section class="banner1">

    <header class="header">
      <div class="container">
        <div class="header_inner">
          <div class="logo">
            <a class="logo_link" href="/mainpage.html"> </a>
          </div>
          <select class="c-Ing">
            <option value="ukr">UKR</option>
            <option value="eng">ENG</option>
          </select>
          <nav class="menu">
            <button onclick="dropdownlistbtn()" class="menu_btn"></button>
            <ul id="drop" class="dropdown">
              <li class="padd">
                <a class="lng-observatory a" href="/observation.php">
                  <strong class="lng-observatory"></strong></a>
              </li>
              <li class="padd">
                <a class="lng-news a" href="/news.php"><strong class="lng-news"></strong></a>
              </li>
              <li class="padd">
                <a class="lng-cosmos a" href="/cosmos.html"><strong class="lng-cosmos"></strong></a>
              </li>
              <li class="padd">
                <a class="lng-events a" href="/events.php"><strong class="lng-events"></strong></a>
              </li>
              <li>
                <a class="iconp" href="/profile.php">
                  <p class="lng-prof"></p>
                  </a>
              </li>
            </ul>
          </nav>
        </div>
      </div>
    </header>
  </section>

```

```

        </ul>
    </nav>
</div>
</div>

</header>
<div class="planet_1"></div>
<div class="img_banner bg">

    <div class="notice">
        <div class="text">
            <h1 class="lng-t1"></h1>
            <p class="lng-t2"></p>
        </div>

    </div>
</div>
</section>
<section class="banner2">
    <div class="container">
        <div class="items">
            <div class="fullb1">
                <div class="b1">
                    <h2 class="lng-t3"></h2>
                    <p class="lng-t4"></p>
                    <a href="profile.php">
                        <strong class="lng-t5"></strong></a>
                </div>

            </div>
            <div class="fullb1">
                <div class="b1">
                    <h2 class="lng-t6"></h2>
                    <p class="lng-t7"></p>
                    <a href="events.php"><strong class="lng-t8"></strong></a>

                </div>
            </div>
        </div>
    </div>
</div>

    <div class="items">
        <div class="fullb1">

            <div class="b1">
                <h2 class="lng-t9"></h2>
                <p class="lng-t10"></p>
                <a href="news.php"><strong class="lng-t11"></strong></a>
            </div>
        </div>
        <div class="fullb1">
            <div class="b1">

                <h2 class="lng-t12"></h2>
                <p class="lng-t13"></p>
                <a href="cosmos.html"><strong class="lng-t14"></strong></a>
            </div>
        </div>
    </div>
</section>
<section class="banner3">
    <div class="container3">

```

```

<div class="banimg">

    <div class="absolute">
        <div class="a3">
            <h1 class="lng-t15"></h1>
        </div>
        <div class="a2">
            <h1 class="lng-t16"></h1>
        </div>
        <div class="a1">
            <h1 class="lng-t17"></h1>
        </div>
        <div class="a2">
            <h1 class="lng-t18"></h1>
        </div></div>
        <canvas id="banner" class="canvstar">
        </canvas>
    </div>
</div>
</section>
<footer class="foot">
    <p class="lng-foot1"></p>
    <p class="lng-foot2"></p>
</footer>
</body>
<script src="/js/lang.js "></script>
<script src="/js/translate.js "></script>
<script type="module" src="/js/starbg.js"></script>

</html>

<?php
include 'connect.php';
session_start();
$user_id = $_SESSION['user_id'];
$news = mysqli_query($db, "SELECT * FROM `news` ORDER BY news_id DESC");

?>
//Сторінка обсерваторії (observation.php)
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="css/reset.css">
    <link rel="stylesheet" href="css/style.css">
    <script src="/js/dropdownlistbtn.js"></script>
    <script type="module" src="/js/suntimes.js"></script>
    <script src="/js/clock.js"></script>

    <script>
    </script>

    <title>LauX</title>
</head>

<body>
<a href="#top"></a>

```

```

<section class="banner1">
  <header class="header">
    <div class="container">
      <div class="header_inner">
        <div class="logo">
          <a class="logo_link" href="/mainpage.html"> </a>
        </div>
        <select class="c-Ing">
          <option value="ukr">UKR</option>
          <option value="eng">ENG</option>
        </select>
        <nav class="menu">
          <button onclick="dropdownlistbtn()" class="menu_btn"></button>
          <ul id="drop" class="dropdown">
            <li class="padd">
              <a class="Ing-observatory a" href="/observation.php">
                <strong class="Ing-observatory"></strong></a>
            </li>
            <li class="padd">
              <a class="Ing-news a" href="/news.php"><strong class="Ing-news"></strong></a>
            </li>
            <li class="padd">
              <a class="Ing-cosmos a" href="/cosmos.html"><strong class="Ing-cosmos"></strong></a>
            </li>
            <li class="padd">
              <a class="Ing-events a" href="/events.php"><strong class="Ing-events"></strong></a>
            </li>
            <li>
              <a class="iconp" href="/profile.php">
                <p class="Ing-prof"></p>
                </a>
            </li>
          </ul>
        </nav>
      </div>
    </div>
  </header>
  <section class="fullp">
    <div class="container2">
      <div class="breadcrumb">
        <ul class="bread">
          <li>
            <a href="/mainpage.html">LauX</a>
          </li>
          /
          <li>
            <a class="Ing-observatory2 a" href="/observation.php">
              <strong class="Ing-observatory2"></strong></a>
          </li>
        </ul>
      <div class="line"></div>
    </div>
    <div class="mainc">
      <div class="textobs">
        <h1 class="Ing-o1"></h1>
        <p class="Ing-o2"></p>
        <h2 class="Ing-o3"></h2>
        <p class="Ing-o4"></p>
        <h2 class="Ing-o5"></h2>
      </div>
    </div>
  </section>

```

```

<p class="lng-o6"></p>
<h2 class="lng-o7"></h2>

<div class="row">
  <div class="ppl">
    
    <p class="lng-researcher1"></p>
  </div>
  <div class="ppl">
    
    <p class="lng-researcher2"></p>
  </div>
  <div class="ppl">
    
    <p class="lng-researcher3"></p>
  </div>
</div>

</div>
</div>
<div class="menubar">
  <div class="menu1">
    <div class="sunrise">
      
      <h3>
        <strong class="lng-sunriset"></strong><strong id="sunrise_time"></strong>
      </h3>
    </div>

    <div class="sunset">
      <h3><strong class="lng-sunsett"></strong><strong id="sunset_time"></strong></h3>
    </div>
  </div>
  <div class="line"></div>

  <div class="menu1">

    <div class="curtime">
      <div id="timezone">
        <p class="lng-time"></p>
      </div>
      <div id="clock"></div>
    </div>
  </div>
  <div class="line"></div>

  <?php
  if($_COOKIE['user'] == ")
  {
?>

  <div class="menu1">
    <div class="loginmin">
      <p class="lng-logoutto"></p>
      <a class="lng-lgnamea" href="profile.php"></a>
    </div>
  </div>
  <?
}else{
  ?>
  <div class="minilog">
  <?

```

```

        $locuser = $_COOKIE['user'];
$locuserinfo = mysqli_query($db, "SELECT * FROM `user` WHERE id = '$locuser'") or die('query failed');
    if(mysqli_num_rows($locuserinfo) > 0){
        $fetch = mysqli_fetch_assoc($locuserinfo);
    }

    if($fetch['image'] == ""){
        echo '';
    }else{
        echo '';
    }

    echo '<a class="lgnamea" href="profile.php"><p class="logname">';
    echo $fetch['name'];
    echo '</p></a>';

?>
</div>
<?
}
?>

```

```

        <div class="line"></div>
    </div>
</div>
</section>
</section>

<footer class="foot ">
    <p>LauX astronomy observatory</p>
    <p>NTU DP 2023 Skitenko A.Y 122-19-4</p>
</footer>
</body>
<script src="/js/lang.js "></script>
<script src="/js/translate.js "></script>
</html>

```

```

//Сторінка новин (news.php)
<?php

include 'connect.php';
session_start();
$user_id = $_SESSION['user_id'];
$news = mysqli_query($db, "SELECT * FROM `news` ORDER BY news_id DESC");

?>
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="css/reset.css">
    <link rel="stylesheet" href="css/style.css">
    <script src="/js/dropdownlistbtn.js"></script>
    <script type="module" src="/js/suntimes.js"></script>
    <script src="/js/clock.js"></script>

```

```

<title>LauX</title>
</head>

<body>

<section class="banner1">

<header class="header">
<div class="container">
<div class="header_inner">
<div class="logo">
<a class="logo_link" href="/mainpage.html"> </a>
</div>
<select class="c-Ing">
<option value="ukr">UKR</option>
<option value="eng">ENG</option>
</select>
<nav class="menu">
<button onclick="dropdownlistbtn()" class="menu_btn"></button>
<ul id="drop" class="dropdown">
<li class="padd">
<a class="Ing-observatory a" href="/observation.php">
<strong class="Ing-observatory"></strong></a>
</li>
<li class="padd">
<a class="Ing-news a" href="/news.php"><strong class="Ing-news"></strong></a>
</li>
<li class="padd">
<a class="Ing-cosmos a" href="/cosmos.html"><strong class="Ing-cosmos"></strong></a>
</li>
<li class="padd">
<a class="Ing-events a" href="/events.php"><strong class="Ing-events"></strong></a>
</li>
<li>
<a class="iconp" href="/profile.php">
<p class="Ing-prof"></p>
</a>
</li>
</ul>
</nav>
</div>
</div>
</header>
<section class="fullp">
<div class="container2">
<div class="breadcrumb">
<ul class="bread">
<li>
<a href="/mainpage.php">LauX</a>
</li>
/
<li>
<a class="Ing-news2 a" href="/news.php"><strong class="Ing-news2" ></strong></a>
</li>
</ul>
<div class="line"></div>
</div>
<div class="mainc">
<div class="wrap">

```



```

<?
while ($fetch2 = mysqli_fetch_assoc($news)) {
echo '<div class="post">';
echo '<div class="iddis">'.$fetch2['news_id'].'</div>';
echo '
<div class="textimg"><h3><a href=newspage.php?id='.$fetch2['news_id'].'#ukr>';
echo $fetch2['title'];
echo '</a></div></h3><p>';
echo $fetch2['date_p'];
echo '</p><a href="">'.$fetch2['author_name'].'</a></div>';
}

?>
</div>

</div>

<div class="menubar">
<div class="menu1">
<div class="sunrise">

<h3>
<strong class="lng-sunriset"></strong><strong id="sunrise_time"></strong>
</h3>
</div>

<div class="sunset">
<h3><strong class="lng-sunsett"></strong><strong id="sunset_time"></strong></h3>
</div>
</div>
<div class="line"></div>

<div class="menu1">

<div class="curtime">
<div id="timezone">
<p class="lng-time"></p>
</div>
<div id="clock">00:00:00</div>

</div>
</div>
<div class="line"></div>
<?php

if($_COOKIE['user'] == "")
{

?>

<div class="menu1">
<div class="loginmin">
<p class="">Вхід не було виконано</p>
<a class="lgnamea" href="profile.php"> Увійти до профілю</a>
</div>
</div>
<?
}else{
?>
<div class="minilog">
<?
$locuser = $_COOKIE['user'];
$locuserinfo = mysqli_query($db, "SELECT * FROM `user` WHERE id = '$locuser'") or die('Перевірьте Ваші
данні...');

```

```

if(mysqli_num_rows($locuserinfo) > 0){
    $fetch = mysqli_fetch_assoc($locuserinfo);
}

if($fetch['image'] == ""){
    echo '';
} else{
    echo '';
}

    echo '<a class="lgnamea" href="profile.php"><p class="logname">';
    echo $fetch['name'];
    echo '</p></a>';
?>
</div>
<?
}
?>

        <div class="line"></div>
    </div>
</div>
</section>
</section>

<footer class="foot">
    <p>LauX astronomy observatory</p>
    <p>NTU DP 2023 Skitenko A.Y 122-19-4</p>
</footer>
</body>
<script src=" /js/lang.js"></script>
<script src="/js/translate.js"></script>
</html>

//сторінка «Про космос» (cosmos.html)
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="css/reset.css">
    <link rel="stylesheet" href="css/style.css">
    <link rel="stylesheet" href="css/form.css">
    <script src="/js/formvalid.js"></script>

    <script type="importmap">
        { "imports": { "three": "https://unpkg.com/three@0.138.0/build/three.module.js", "OrbitControls":
"https://unpkg.com/three@0.138.0/examples/jsm/controls/OrbitControls.js" } }
    </script>
    <script src="/js/dropdownlistbtn.js"></script>

    <title>LauX</title>
</head>

<body>

    <section class="banner1">

        <header class="header">
            <div class="container">

```

```

<div class="header_inner">
  <div class="logo">
    <a class="logo_link" href="/mainpage.html"> </a>
  </div>
  <select class="c-Ing">
    <option value="ukr">UKR</option>
    <option value="eng">ENG</option>
  </select>
  <nav class="menu">
    <button onclick="dropdownlistbtn()" class="menu_btn"></button>
    <ul id="drop" class="dropdown">
      <li class="padd">
        <a class="Ing-observatory a" href="/observation.php">
          <strong class="Ing-observatory"></strong></a>
      </li>
      <li class="padd">
        <a class="Ing-news a" href="/news.php"><strong class="Ing-news"></strong></a>
      </li>
      <li class="padd">
        <a class="Ing-cosmos a" href="/cosmos.html"><strong class="Ing-cosmos"></strong></a>
      </li>
      <li class="padd">
        <a class="Ing-events a" href="/events.php"><strong class="Ing-events"></strong></a>
      </li>
      <li>
        <a class="iconp" href="/profile.php">
          <p>Профіль</p>
          </a>
        </li>
    </ul>
  </nav>
</div>

</div>
</header>
<section class="fullp">
  <div class="cosmoscontent">
    <div class="constellations">
      <div>
        
        <p class="Ing-ursamajor"></p>
      </div>
      <div>
        
        <p class="Ing-ursaninor"></p>
      </div>
    </div>
  </div>

  <div class="planets">
    <div class="mercury"></div>
    <div class="planetpop" id="mercury">
      <div class="closemer">X</div>
      <div class="row between">
        <canvas id="mercuryholder"></canvas>
        <div class="text">
          <h2 class="Ing-mercury"></h2>
          <p class="Ing-mer1"></p>
          <p class="Ing-mer2"></p>
          <p class="Ing-mer3"></p>
          <p class="Ing-mer4"></p>
        </div>
      </div>
    </div>
  </div>

```

```

</div>
<div class="lng-merinfo">
  <p class="lng-mer5"></p>
  <p class="lng-mer6"></p>
</div>
</div>
<div class="venus"></div>
<div class="planetpop" id="venus">
  <div class="closeven">X</div>
  <div class="row between">
    <canvas id="venusholder"></canvas>
    <div class="text">
      <h2 class="lng-venus"></h2>
      <p class="lng-ven1"><b></p>
      <p class="lng-ven2"></p>
      <p class="lng-ven3"></p>
      <p class="lng-ven4"></p>
    </div>
  </div>
  <div class="info">
    <p class="lng-mer5"></p>
    <p class="lng-ven5"></p>
  </div>

</div>
<div class="earth"></div>
<div class="planetpop" id="earth">
  <div class="closeear">X</div>
  <div class="row between">
    <canvas id="earthholder"></canvas>
    <div class="text">
      <h2 class="lng-earth"></h2>
      <p class="lng-ear1"></p>
      <p class="lng-ear2"></p>
      <p class="lng-ear3"></p>
      <p class="lng-ear4"></p>
    </div>
  </div>
  <div class="info">
    <p class="lng-ear5"></p>
    <p class="lng-ear6"></p>
  </div>
</div>
<div class="mars"></div>
<div class="planetpop" id="mars">
  <div class="closemar">X</div>
  <div class="row between">
    <canvas id="marsholder"></canvas>
    <div class="text">
      <h2 class="lng-mars"></h2>
      <p class="lng-mar1"></p>
      <p class="lng-mar2"></p>
      <p class="lng-mar3"></p>
      <p class="lng-mar4"></p>
    </div>
  </div>
  <div class="info">
    <p class="lng-mar5"></p>
    <p class="lng-mar6"></p>
  </div>
</div>
<div class="jupiter"></div>
<div class="planetpop" id="jupiter">

```

```

<div class="closejup">X</div>
<div class="row between">
  <canvas id="jupiterholder"></canvas>
  <div class="text">
    <h2 class="lng-jupiter"></h2>
    <p class="lng-jup1"></p>
    <p class="lng-jup2"></p>
    <p class="lng-jup3"></p>
    <p class="lng-jup4"></p>
  </div>
</div>
<div class="info">
  <p class="lng-jup5"></p>
  <p class="lng-jup6"></p>
</div>

</div>

</div>
<div class="constellations">
  <div>
    <p class="lng-cassiopeia"></p>
    
  </div>
  <div>
    <p class="lng-cenes"></p>
    
  </div>
  <div>
    <p class="lng-cepheus"></p>
    </div>
</div>
</div>

</section>
<section> <canvas id="fullrender"></canvas></section>
</section>

<footer class="foot">
  <p class="lng-foot1"></p>
  <p class="lng-foot2"></p>
</footer>
</body>
<script>
  let close1 = document.querySelector(".closemer");
  let close2 = document.querySelector(".closeven");
  let close3 = document.querySelector(".closeear");
  let close4 = document.querySelector(".closeemar");
  let close5 = document.querySelector(".closejup");

  let mercury = document.querySelector(".mercury");
  let popmer = document.getElementById("mercury");
  let venus = document.querySelector(".venus");
  let popven = document.getElementById("venus");
  let earth = document.querySelector(".earth");
  let popear = document.getElementById("earth");
  let mars = document.querySelector(".mars");
  let popmar = document.getElementById("mars");
  let jupiter = document.querySelector(".jupiter");
  let popjup = document.getElementById("jupiter");

  mercury.addEventListener("click", () => {
    popmer.style.display = "flex";

```

```

    })
    venus.addEventListener("click", () => {
        popven.style.display = "flex";
    })
    earth.addEventListener("click", () => {
        popear.style.display = "flex";
    })
    mars.addEventListener("click", () => {
        popmar.style.display = "flex";
    })
    jupiter.addEventListener("click", () => {
        popjup.style.display = "flex";
    })

    close1.addEventListener("click", () => {
        popmer.style.display = "none";
    })
    close2.addEventListener("click", () => {
        popven.style.display = "none";
    })
    close3.addEventListener("click", () => {
        popear.style.display = "none";
    })
    close4.addEventListener("click", () => {
        popmar.style.display = "none";
    })
    close5.addEventListener("click", () => {
        popjup.style.display = "none";
    })
</script>

<script src="/js/lang.js "></script>
<script src="/js/translate.js "></script>
<script type="module" src="/js/3dscene.js"></script>
<script type="module" src="/js/solrsystembanner.js"></script>

```

```
</html>
```

```
//Сторінка «Події» (events.php)
```

```
<?php
```

```

include 'connect.php';
session_start();
$user_id = $_SESSION['user_id'];
$news = mysqli_query($db, "SELECT * FROM `news` ORDER BY news_id DESC");

```

```
?>
```

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```

<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">

```

```

<link rel="stylesheet" href="css/reset.css">
<link rel="stylesheet" href="css/style.css">
<script src="/js/dropdownlistbtn.js"></script>
<script type="module" src="/js/suntimes.js"></script>
<script src="/js/clock.js"></script>

```

```

<title>LauX</title>
</head>

<body>

<section class="banner1">

<header class="header">
<div class="container">
<div class="header_inner">
<div class="logo">
<a class="logo_link" href="/mainpage.html"> </a>
</div>
<select class="c-Ing">
<option value="ukr">UKR</option>
<option value="eng">ENG</option>
</select>
<nav class="menu">
<button onclick="dropdownlistbtn()" class="menu_btn"></button>
<ul id="drop" class="dropdown">
<li class="padd">
<a class="Ing-observatory a" href="/observation.php">
<strong class="Ing-observatory"></strong></a>
</li>
<li class="padd">
<a class="Ing-news a" href="/news.php"><strong class="Ing-news"></strong></a>
</li>
<li class="padd">
<a class="Ing-cosmos a" href="/cosmos.html"><strong class="Ing-cosmos"></strong></a>
</li>
<li class="padd">
<a class="Ing-events a" href="/events.php"><strong class="Ing-events"></strong></a>
</li>
<li>
<a class="iconp" href="/profile.php">
<p class="Ing-prof"></p>
</a>
</li>
</ul>
</nav>
</div>
</div>
</header>
<section class="fullp">
<div class="container2">
<div class="breadcrumb">
<ul class="bread">
<li>
<a href="/mainpage.html">LauX</a>
</li>
/
<li>
<a class="Ing-events2 a" href="/events.php"><strong class="Ing- events2 " ></strong></a>
</li>
</ul>
<div class="line"></div>
</div>
<div class="mainc">
<div class="wrap">

```

```

<?
while ($fetch2 = mysqli_fetch_assoc($news)) {
echo '<div class="post">';
echo '<div class="iddis">'. $fetch2['news_id']. '</div>';
echo '
<div class="textimg"><h3><a href=newspage.php?id='. $fetch2['news_id']. '#ukr>';
echo $fetch2['title'];
echo '</a></div></h3><p>';
echo $fetch2['date_p'];
echo '</p><a href="">'. $fetch2['author_name']. '</a></div>';
}

?>
</div>

</div>

<div class="menubar">
<div class="menu1">
<div class="sunrise">

<h3>
<strong class="lng-sunriset"></strong><strong id="sunrise_time"></strong>
</h3>
</div>

<div class="sunset">
<h3><strong class="lng-sunsett"></strong><strong id="sunset_time"></strong></h3>
</div>
</div>
<div class="line"></div>

<div class="menu1">

<div class="curtime">
<div id="timezone">
<p class="lng-time"></p>
</div>
<div id="clock">00:00:00</div>

</div>
</div>
<div class="line"></div>
<?php

if($_COOKIE['user'] == "")
{

?>

<div class="menu1">
<div class="loginmin">
<p class="">Вхід не було виконано</p>
<a class="lgnamea" href="profile.php"> Увійти до профілю</a>
</div>
</div>
<?
}else{
?>
<div class="minilog">
<?
$locuser = $_COOKIE['user'];
$locuserinfo = mysqli_query($db, "SELECT * FROM `user` WHERE id = '$locuser'") or die('Перевірьте Ваші
данні...');

```



```

if(mysql_num_rows($locuserinfo) > 0){
    $fetch = mysqli_fetch_assoc($locuserinfo);
}

if($fetch['image'] == ""){
    echo '';
} else{
    echo '';
}

    echo '<a class="lgnamea" href="profile.php"><p class="logname">';
    echo $fetch['name'];
    echo '</p></a>';
?>
</div>
<?
}
?>

        <div class="line"></div>
    </div>
</div>
</section>
</section>

<footer class="foot">
<p class="lng-foot1"></p>
    <p class="lng-foot2"></p>
</footer>
</body>
<script src=" /js/lang.js"></script>
<script src="/js/translate.js"></script>
</html>

```

ВІДГУК

**керівника економічного розділу
на кваліфікаційну роботу бакалавра**

на тему:

**“Розробка веб-орієнтованого застосунку для
астрономічної обсерваторії засобами мов HTML, CSS, JavaScript та PHP”
студентки групи 122-19-4 Скітенко Анастасії Юріївни**

**Керівник економічного розділу
зав.каф. проф. д.е.н.**

О. Г. Вагонова

ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ

Ім'я файлу	Опис
Пояснювальні документи	
кваліфікаційна_робота_бакал авра_ст_гр_122-19- 4 Скітенко А Ю.docx	Пояснювальна записка до кваліфікаційної роботи. Документ Word.
кваліфікаційна_робота_бакал авра_ст_гр_122-19- 4 Скітенко А Ю.pdf	Пояснювальна записка до кваліфікаційної роботи в форматі PDF
Програма	
Скітенко_А_Ю.zip	Архів. Містить коди програми та архів встановлення веб-додатку
Презентація	
презентація_кваліфікаційної_ роботи_ст_гр_122-19- 4 Скітенко А Ю.pptx	Презентація кваліфікаційної роботи