

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 *Філіпюка Івана Володимировича*
(ПІБ)

академічної групи *122-18-3*
(шифр)

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

освітньої програми *Комп'ютерні науки*
(назва освітньої програми)

на тему: *Розробка веб-орієнтованої інформаційної системи для організації продажу онлайн-курсів з програмування*

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

Дніпро
2022

Міністерство освіти і науки України
НТУ «Дніпровська політехніка»

ЗАТВЕРДЖЕНО:

завідувач кафедри
програмного забезпечення комп'ютерних
систем

(повна назва)

І.М. Удовик
(підпис) (прізвище, ініціали)

« » 2022 року

**ЗАВДАННЯ
на кваліфікаційну роботу**

бакалавра

(назва освітньо-кваліфікаційного рівня)

студента 122-18-3 Філіпюка Івана Володимировича
(група) (прізвище та ініціали)

тема кваліфікаційної роботи Розробка веб-орієнтованої інформаційної системи для організації продажу онлайн-курсів з програмування

затверджена наказом ректора НТУ «ДП» від «18» травня 2022 р. № 268-с

Розділ	Зміст виконання	Термін виконання
<i>Спеціальний</i>	<i>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</i>	13.05.2022 р.
<i>Економічний</i>	<i>Провести розрахунок трудомісткості розробки програмного забезпечення, витрат на створення ПЗ й тривалості його розробки</i>	27.05.2022 р.

Завдання видав _____ доц. Спірінцев В.В
(підпис) (посада, прізвище, ініціали)

Завдання прийняв до виконання _____ Філіпюк І.В.
(підпис) (прізвище, ініціали)

Дата видачі завдання:

Термін подання кваліфікаційної роботи до ЕК:

РЕФЕРАТ

Пояснювальна записка: 93 с., 18 рис., 3 дод., 24 джерела.

Об'єкт розробки: система функціонування сайту продажу онлайн-курсів з програмування.

Мета кваліфікаційної роботи: розробка веб-орієнтованої інформаційної системи для організації продажу онлайн-курсів з програмування.

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

У першому розділі кваліфікаційної роботи розглянуто загальні відомості з предметної галузі, описано сучасний стан проблеми, описані переваги які надає використання розроблюваної інформаційної системи, та основні вимоги до неї. Також в цьому розділі описано завдання до кваліфікаційної роботи, його мета, та доцільність його рішення.

У другому розділі розглянуто опис використаних технологій для створення інформаційної системи, функціональне призначення системи, опис структури системи та алгоритмів її функціонування. Детально розглянуто інтерфейс системи та її можливості.

В економічному розділі знаходяться розрахунки трудомісткості та вартості розробленого програмного продукту, та визначено приблизний час на його виконання.

Список ключових слів: САЙТ, ВЕБ-РЕСУРС, РОЗРОБКА, UI ДИЗАЙН, UX ДИЗАЙН, ІНТЕРНЕТ, ОНЛАЙН-КУРСИ.

ABSTRACT

Explanatory note: 93 pp., 18 figs., 3 appendices, 24 sources.

Object of development: the system of operation of the site for the sale of online programming courses.

The purpose of the qualification work: Development of a web-based information system for organizing the sale of online programming courses.

The introduction considers the analysis and the current state of the problem, specifies the purpose of the qualification work and the field of its application, provides a justification for the relevance of the topic and clarifies the task.

The first section of the qualification work considers general information on the subject area, describes the current state of the problem, describes the benefits of using the developed information system, and the basic requirements for it. Also in this section describes the tasks for the qualification work, its purpose, and the feasibility of its solution. The second section describes the technologies used to create an information system, the functional purpose of the system, a description of the structure of the system and algorithms for its operation. The system interface and its capabilities are considered in detail.

The economic section contains calculations of the complexity and cost of the developed software product, and determines the approximate time for its implementation.

List of keywords: WEBSITE, WEB RESOURCE, DEVELOPMENT, UI DESIGN, UX DESIGN, INTERNET, ONLINE COURSES.

ЗМІСТ

РЕФЕРАТ.....	3
ABSTRACT.....	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	7
ВСТУП.....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ.....	10
1.1.1 Загальні відомості з предметної галузі.....	10
1.1.2 Класифікація сайтів.....	12
1.1.3 Електронна комерція.....	15
1.2. Призначення розробки та галузь застосування.....	21
1.3. Підстава для розробки.....	22
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.....	23
1.5.3. Вимоги до складу та параметрів технічних засобів.....	26
1.5.4. Вимоги до інформаційної та програмної сумісності.....	26
РОЗДІЛ 2. ПРОЕКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	27
2.1. Функціональне призначення системи.....	27
2.2. Опис застосованих математичних методів.....	28
2.3. Опис використаних технологій та мов програмування.....	28
2.4. Опис структури системи та алгоритмів її функціонування.....	37
2.5. Обґрунтування та організація вхідних та вихідних даних програми.....	42
2.6. Опис роботи розробленої системи.....	42

2.6.1.	Використані технічні засоби.....	42
2.6.2.	Використані програмні засоби.....	43
2.6.3.	Виклик та завантаження програми.....	44
2.6.4.	Опис інтерфейсу користувача.....	44
РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ.....		52
3.1.	Розрахунок трудомісткості та вартості розробки програмного продукту.....	52
3.2.	Розрахунок витрат на створення програми.....	56
ВИСНОВКИ.....		59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....		60
Додаток А. Код програми.....		63
Додаток Б. Відгук керівника економічного розділу.....		92
Додаток В. Перелік файлів на диску.....		93

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД -	База даних
CSS -	Cascading Style Sheets
IT -	Інформаційні технології
FE -	FrontEnd
JS -	Java Script
HTML -	Hypertext Markup Language
HTTP -	Hypertext Transfer Protocol
FE -	FrontEnd
BE -	BackEnd
UI -	User Interface
UX -	User Experience

ВСТУП

Освіта – це інструмент, який дає людям знання, вміння, інформацію, дозволяє їм знати свої права та обов’язки щодо своєї сім’ї, суспільства, нації. Освіта збагачує світогляд, розширює можливості боротьби з несправедливістю, насильством, корупцією та багатьма іншими неприємними моментами в житті суспільства. Освіта дає нам знання про навколишній світ, які розвивають прогресивні погляди на життя. А це найголовніше в еволюції нації. Без освіти ніхто не буде досліджувати нові ідеї. Це означає, що людина не зможе розвивати світ, тому що без ідей немає творчості, а без творчості немає розвитку нації. Освіта є важливим аспектом, який відіграє важливу роль у сучасному індустріальному світі. Людям потрібна хороша освіта, щоб мати можливість вижити в конкурентному світі. Освіта допомагає усунути бідність. Адже, коли людина здобула освіту, вона може влаштуватися на хорошу роботу та задовольнити свої потреби та потреби родини на належному рівні.

Освіта дуже важлива річ в нашому житті. Це можливість для людини стати не просто кимось, а мати знання. Знання це головне. Це головний ресурс і головна валюта у світі. Той, хто володіє інформацією, володіє всесвітом. На мою думку освіта надає людині силу, мужність, рішучість і вплив на своє майбутнє, це найважливіший вплив на життя кожної людини.

Знання забезпечують гарну платформу, з якої ви можете оцінити ваші здібності у тій чи іншій галузі, а відтак - сформувати навички, які потенційно приведуть вас до професійного успіху. Особливо це стає важливо з віком, коли у результаті практичної діяльності ми маємо вже сформовані навички та діємо за завченими парадигмами, закладеними почасти ще в юності, під час здобуття першої освіти. Та світ навколо нас змінюється і ми просто зобов’язані змінюватись і застосовувати плоди людського прогресу.

Але зараз все складніше вчитись, через безліч обставин. Деяка інформація, яку надають в навчальних закладах, вже може бути не

актуальною. Навіть цілі професії перестають мати актуальність через те, що світ не стоїть на місці. Все рухається, людство не стоїть на місці і постійно вдосконалює різні процеси, професії. А ще багато інших факторів на це впливають. Зараз всі переживають дуже складний етап. Ніхто не міг подумати, що на їх долю випаде пандемія і війна. Для всіх це складно. Але це ще більше показує, що в такому непостійному світі треба шукати можливості для навчання і не зупинятись.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Загальні відомості з предметної галузі

Почалась ера нових інформаційних та комунікаційних технологій, яка принесла значні зміни у сферу виробництва, освіти та ділової активності людей. Суспільство, в якому ми живемо, набуває все більше рис інформаційного і це позначається у всіх сферах життя. Сферу освіти можна назвати переднім флангом, який приймає основний удар наступу інформаційних технологій.

Усі сходяться на думці, що Інтернет стимулює процес розвитку інтелектуальних здібностей, надаючи багато додаткової інформації.

Водночас наголошується, що Інтернет є лише засобом у навчальному процесі та використання Інтернет-технологій має повністю контролюватись вчителем, при цьому до вчителя пред'являються такі вимоги, як комп'ютерна грамотність та знання Інтернет-технологій. Вчителів не може не хвилювати проблема підвищення своєї кваліфікації та, отже, рівня освіти загалом. Настав час переосмислення, розширення традиційних напрямів навчання, пошук нових методів та прийомів зацікавити дітей до навчання. Таким чином, сьогодні є актуальним питання про необхідність та перспективи створення сайту з надання послуг освіти.

Також не треба ігнорувати матеріальну частину вопросу, так як на даний момент все більше людей думають про альтернативні способи отримання особливої інформації в цій галузі. Тож розробка сайтів з надання послуг освіти, є актуальною, як ніколи.

Веб-сайт (веб-сайт) - це сукупність веб-сторінок та інших ресурсів, з'єднаних за змістом, пов'язаних гіперпосиланнями і фізично що знаходяться на одному сервері. Веб-сайт – це єдиний інформаційний блок.

Мережа Інтернет складається з величезної кількості з'єднаних між собою комп'ютерів, маршрутизаторів та іншого, необхідного для правильної роботи. Кожен елемент мережі Інтернет (вузол) має унікальний описувач - IP-адресу. Знаючи IP-адресу вузла можна спробувати підключитися до нього, а маючи невеликі навички можна визначити комусь ця адреса належить і в якому регіоні світу знаходиться. IP-адреси прийнято записувати у вигляді чотирьох груп цифр, розділених точками, наприклад .168.100.003 або 10.10 .0.123

Запам'ятати адреси всіх часто відвідуваних сторінок практично неможливо, тому в Інтернеті існують спеціальні сервери DNS (Domain Name Resolution), на яких зберігаються списки зіставлення IP-адрес і символічних імен. Саме завдяки цим серверам користувач завжди потрапляє за потрібною IP-адресою, набравши в браузері тільки ім'я сторінки.

Після того, як ми ввели в рядок браузера ім'я потрібної сторінки, браузер самостійно отримує з DNS IP-адресу потрібного сервера і надсилає на цю адресу спеціальний запит на отримання сторінки (HTTP-запит). Спеціалізована програма (т.зв. Web-сервер), що працює на сервері, обробляє цей запит і повертає в браузер необхідну сторінку.

Очевидно, що всі дії відображення сторінки можна однозначно розділити на дві категорії: виконувані на стороні клієнта (клієнтський код або FE) і на стороні сервера (серверний код або BE). Причому сервер нічого не знає про поточний стан клієнта, а клієнт про поточний стан сервера. При розробці алгоритмів обміну треба завжди пам'ятати про це і своєчасно передавати необхідні дані, що описують стан або необхідну дію.

Залежно від місця застосування різняться кошти реалізації елементів. На стороні клієнта зазвичай використовується тільки HTML, JS (AJAX), CSS і Flash.

1 .1.1. Класифікація сайтів

Сайт візитка

Сайт візитка - зазвичай це сайт з кількох сторінок , на якому міститься інформація про діяльність фірми чи приватної особи, послуги і контактні дані для потенційних клієнтів. Також за бажанням можна розмістити історію фірми, види діяльності, заслуги, фотографії тощо. Такий сайт являється повноцінним віртуальним ресурсом з унікальним дизайном та оформленням.

Корпоративний сайт

Корпоративний сайт є повноцінним представленням Вашої компанії в інтернеті. Це обов'язковий інструмент що допомагає забезпечити успішне ведення бізнесу будь-якої компанії . Мета створення корпоративного сайту – довести інформацію про компанію, її діяльність, товари або послуги, до відома якомога більшої кількості людей, в яких зацікавлена компанія як в майбутніх споживачах або партнерах.

Сайт каталог (вітрина)

Призначений для продажу товарів чи групи товарів. На таких сторінках потрібно розмістити всю інформацію, необхідну для того , щоб клієнт побачив, зацікавився і купив даний товар. Це структурований каталог продукції Вашої компанії . На таких сайтах розміщується інформація про товари /послуги, яку неможливо помістити в прайс-лист , детальний опис, фото, ціна.

Сайти електронної комерції (інтернет магазини)

На відміну від сайту-вітрини , де клієнт може переглянути наявність всіх товарів , за допомогою інтернет-магазину клієнт може ще й зробити замовлення, вибрати варіант розрахунку, спосіб отримання замовлення та одержати рахунок на оплату . І все це не відходячи від власного комп'ютера.

Інформаційні портали та сервіси

Інформаційний портал - це веб-сайт, який об'єднує в собі безліч різних

сервісів, і надає користувачеві мережі Інтернет доступ до них відповідно обраного тематичного розділу порталу, об'ємний, він має велику кількість інформації, і може запропонувати велику кількість послуг. Інформаційний портал є засіб збору, обробки і розповсюдження інформації з метою рішення конкретних завдань. Тут користувач може зареєструватися і отримати доступ до всіх послуг цього вузла — електронної пошти, блогів, пошуку роботи, служби оголошень, погоди, новин, гороскопів, а також, відділу торгівлі, подорожей, відсилання SMS та багато іншого.

Промо-сайт

Такий сайт це насамперед рекламний інструмент, для реклами конкретного товару чи послуги чи навіть якогось заходу. На промо-сайтах розміщується вичерпна інформація про бренд, різні рекламні акції (конкурси, вікторини, ігри тощо). Не великий за об'ємом текстового наповнення, повинен володіти яскравим, незабутнім, стильним дизайном, що поєднує велику кількість графіки і здатним ефектно проінформувати користувача про товар чи послугу.

Сайт-брошура

Сайти-брошури є найпоширенішими, вони містять практично ті ж матеріали, що й традиційні рекламні брошури. Їхньою метою є рекламування послуг та товарів певної компанії.

Особистий сайт

Назва говорить сама за себе - особистий сайт. Це може бути що завгодно. Розкажи про себе цілому світові, пиши свої статті, історії, розміщуй фото, відео тощо. Часто створюють особистий сайт з метою ведення блогу, інформування про свою подорож онлайн, презентацію власних робіт тощо. Інколи особисті сайти створюються без попередньо визначеної мети, просто так, для себе, а з часом, при наповненні контентом, можуть перетворитися в потужний інтернет ресурс.

Блог - це веб-сайт, головний зміст якого записи, зображення чи мультимедіа, що регулярно додаються. Для блогів характерні короткі записи

тимчасової значущості.

Інші веб-сайти

Це найбільш поширені категорії інтернет сторінок (веб-сторінок) які найчастіше створюються на основі заготовок, шаблонів. Звичайно існує багато інших видів сторінок, таких як розважальних, інформаційних, всілякі форуми і інші.

Технологія відображення ресурсу:

- статичні сайти - складаються з HTML файлів і передаються користувачеві у вигляді , в якому зберігаються на сервері;
- динамічні сайти - як правило, працюють на основі системи управління сайтом (CMS). Вміст генерується, за допомогою різних скриптів використовуючи базу даних.

Сайт візитка -представництво компанії в мережі Інтернет. Надає загальну інформацію про власника (компанія, підприємець і тд). Як правило, складається з 5-7 сторінок, при цьому максимально описує вид діяльності. Головні розділи - про компанію, прайс-листи, контактні дані, схема проїзду.

Бізнес сайт -це каталог продукції компанії (товари або послуги). У каталозі присутній докладний опис товару, публікуються сертифікати й нагороди, відгуки клієнтів і звичайно технічні дані. Сайт даної специфікації незамінний для будь-якої компанії.

Промо сайт - веб сайт конкретної торгової марки (бренду). Створюється з метою реклами нового продукту або бренду в цілому. Розділи: акції, вікторини, голосування, ігри і тд.

Інтернет - магазин це каталог товарів з можливістю онлайн замовлення. Клієнт має можливість оплати товару, за допомогою різних сервісів використовуючи при цьому пластикову карту, рахунок або Інтернет гроші. Розділи: про магазин, допомога, каталог, кошик, сторінка замовлення, контакти.

Корпоративний сайт - містить повну інформацію про компанію (послуги, товари і тд.). Як правило, має складну структуру і має розширену

функціональність. Може бути відкритий для відвідувача або призначений для внутрішнього використання (зв'язок , документообіг і тд .).

1.1.2. Електронна комерція

Електронна комерція, або e-commerce - це сфера економіки, коли торгіві і фінансові операції проводяться в інтернеті. Якщо говорити простими словами , це будь-яка транзакція, здійснена з електронного пристрою, підключеного до мережі. Аналог торгового центру , але з великим асортиментом і комфортом: його можна відвідати, не виходячи з дому.

Інтернет надає унікальні можливості для комунікації з аудиторією . Спочатку їх гідно оцінили в США, запустивши електронну торгівлю в 1979, потім перспективний напрям поширився по Європі, починаючи з 1981, і вже в кінці 1990-х Китай та країни колишнього Радянського Союзу підхопили естафету.

Цифрова економіка щомісяця набирає обертів: приблизно 8 млрд чоловік населяють нашу планету, до інтернету же підключено 7 млрд пристроїв. Тепер електронній комерції відведена значна частина світового фінансового ринку.

Висновок можна зробити наступний: зі зростанням поширення інтернету , конкурентоспроможними будуть тільки ті компанії, у яких є своя освоєна ніша в цифровому світі .

E-commerce в Україні - це поширення, продаж , реклама, просування послуг і товарів, тобто будь -які угоди у всесвітній мережі з використанням цифрових пристроїв.

E-commerce ділиться на кілька загальновизнаних категорій. Ось деякі, з якими ми найчастіше зустрічаємося в повсякденному житті:

- C2C (Consumer-to-Consumer). Схема «споживач-до-споживача». У приклад можна привести торговий майданчик OLX, eBay, і подібні до них, де людина, навіть якщо вона не підприємець, може виставити на продаж якусь річ;
- B2B (Business-to-Business). Схема «бізнес-до-бізнесу». Її характеризує продаж оптових партій товару від виробника дилера. Дилер віддає товар дрібним оптом інтернет-магазинам, звідки він надходить кінцевому споживачеві. Власне, це і є третя головна категорія e-trade;
- B2C (Business-to-Consumer), «бізнес-споживач». Розрахунки між онлайн-магазином і клієнтом, покупка навчальних курсів у зареєстрованих експертів, оренда програмного забезпечення - будь - яка роздрібна угода між юридичними і фізичними особами;
- E2E (Exchange-to-Exchange), «біржа-до -біржі». В Інтернеті термін E2E використовується для позначення обміну інформацією або угод між веб -сайтами, які самі виступають в якості бірж або брокерів для обміну товарами і послугами між підприємствами. Простий приклад: транзакція з електронного гаманця на банківську карту. E2E можна розглядати як форму B2B;
- G2C (Government-to-Citizens), «уряд-до-громадян». Сюди можна віднести сплату податків, комунальних послуг, ліцензій, і так далі . Також громадяни отримують необхідну державну інформацію, що вже не можна назвати електронною комерцією.

Є й інші схеми, в основному, це вузькі відгалуження вже перерахованих категорій.

Переваги e-commerce

Основний, але далеко не єдиний, плюс інтернет -торгівлі полягає в відсутності географічних обмежень. Співпраця з логістичними компаніями допоможе охопити аудиторію в світовому масштабі. Так діють гіганти інтернет-торгівлі , такі як AliExpress, Alibaba, eBay, Amazon, Zappos, iTunes.

Коли бренд стає популярний, логотип пізнаваний , бізнес отримує нових користувачів по «сарафанному» радіо : люди діляться посиланнями в соціальних мережах на цікаві товари, приємні ціни або УТП (унікальна торгова пропозиція). Таким чином, інтернет допомагає привести умовно безкоштовних клієнтів.

Торговельне обладнання, найм персоналу , оренда приміщень - це все для торгівлі офлайн . В інтернеті про ці витрати можна забути .

Пошук цільової аудиторії став легше - людям подобається збиратися в групи за інтересами в соцмережах , на тематичних форумах і сайтах.

Ціна товару в інтернет-магазині найчастіше нижче, ніж в магазинах офлайн. Це пов'язано зі скороченням ланцюжка посередників між постачальником і замовником, і зменшенням витрат на утримання.

Плюси e-commerce для організацій:

- Зростає якість обслуговування клієнтів;
- Бізнес-процеси стають швидше, ефективніше і простіше;
- Набагато менше паперової роботи;
- Підвищується продуктивність організації: процес запускається , коли надходить запит від клієнта - ні пізніше , ні раніше, без зайвих дій і затримок .

Переваги онлайн-торгівлі для клієнта:

- Підтримка і доступність з будь-якого місця цілодобово: вибрати товар, оформити покупку і доставку можна в будь-який час доби, з будь-якого місця, де є підключення до мережі;
- Можливість вибору і порівняння товарів в різних магазинах ;
- Можна залишити свій коментар або побачити відгуки про товар інших покупців перш, ніж прийняти рішення про покупку;
- Завжди доступна докладна інформація про товар або послугу;

- Підприємець змушений через високу конкуренцію в електронній комерції робити знижки, проводити акції і конкурси. Багато банків пропонують кешбек при оплаті картою.

Незважаючи на сильні плюси, у цифрових фінансових відносинах є і недоліки.

Недоліки онлайн-комерції

Ось головні з них:

- Велика конкуренція в популярних напрямках онлайн-торгівлі. Наприклад, продаючи побутову, відео та аудіотехніку, мобільні телефони доведеться буквально боротися за кожного покупця. В цьому випадку необхідні чималі витрати на маркетинг, SMM просування, SEO оптимізацію сайту;
- Мало довіри до нових інтернет-магазинів з боку користувачів. Цьому сприяли шахраї, які давно звернули увагу на електронну комерцію;
- Відсутність знань про роботу в мережі. Тут працюють принципи, відмінні від торгівлі офлайн: аналітика, логістика, обробка замовлень відбуваються інакше. Затримка з відповідями на запити покупців може зіпсувати імідж компанії;
- Успішний бізнес залежить від стабільного швидкого інтернету і щоденної залученості власника в управління справою;
- Для регулювання бізнес-процесів в мережі потрібна грамотно розроблена законодавча база. Її немає. В тому числі, це стосується захисту персональних даних. Необхідна інтеграція актуального програмного забезпечення, часто дорогого;
- Замовник не може прийти і самостійно забрати покупку. Потрібен ідеально налагоджений механізм доставки;
- Відповідальних віддалених співробітників важко знайти і навчити.

Всі ці проблеми електронної комерції можна вирішити. Звичайно,

доведеться потрудитися.

Мета будь-якого бізнесу - заробити, і цифрова платформа дозволяє зробити це з мінімальними вкладеннями на старті. Звичайно, якщо ніша обрана правильно.

Розглянемо найпопулярніші сервіси з продажу онлайн курсів.

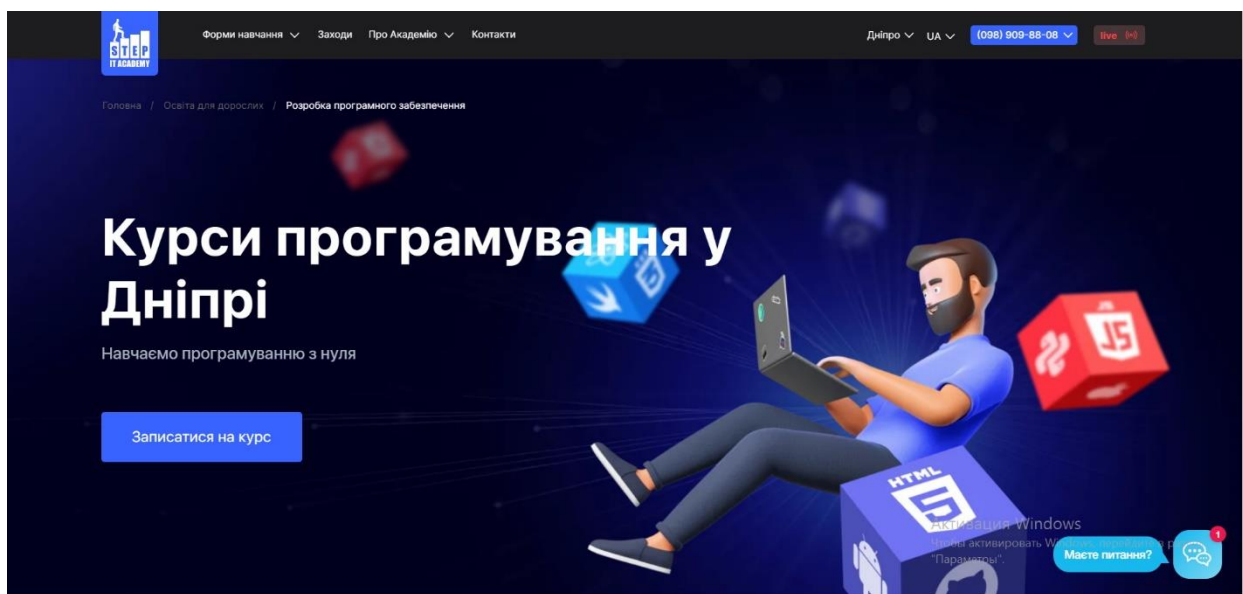


Рис. 1.1. Сайт онлайн школи Степ

Онлайн школа Степ (рис.1.1) [23]. Маємо головну сторінку з яскравим дизайном , меню і великими літерами, які нагадують клієнту для чого він натиснув на сайт. Але в цього сайта є великий недолік - дуже багато повідомлень які відволікають і відлякують (рис .1.2).

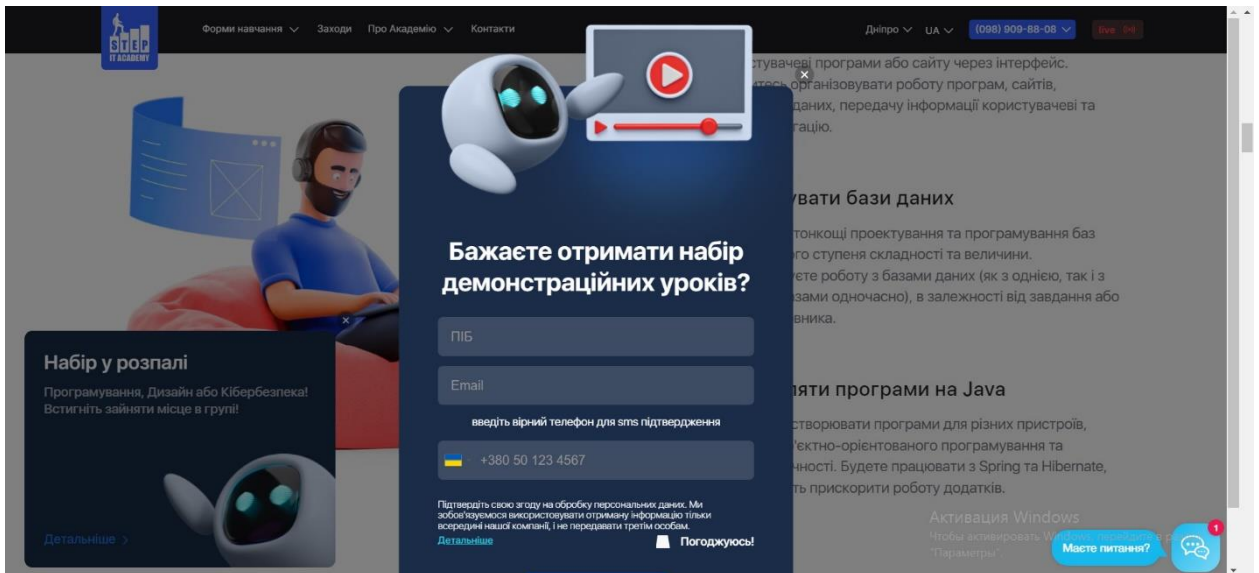


Рис. 1.2. Спливаючі повідомлення сайту Степ

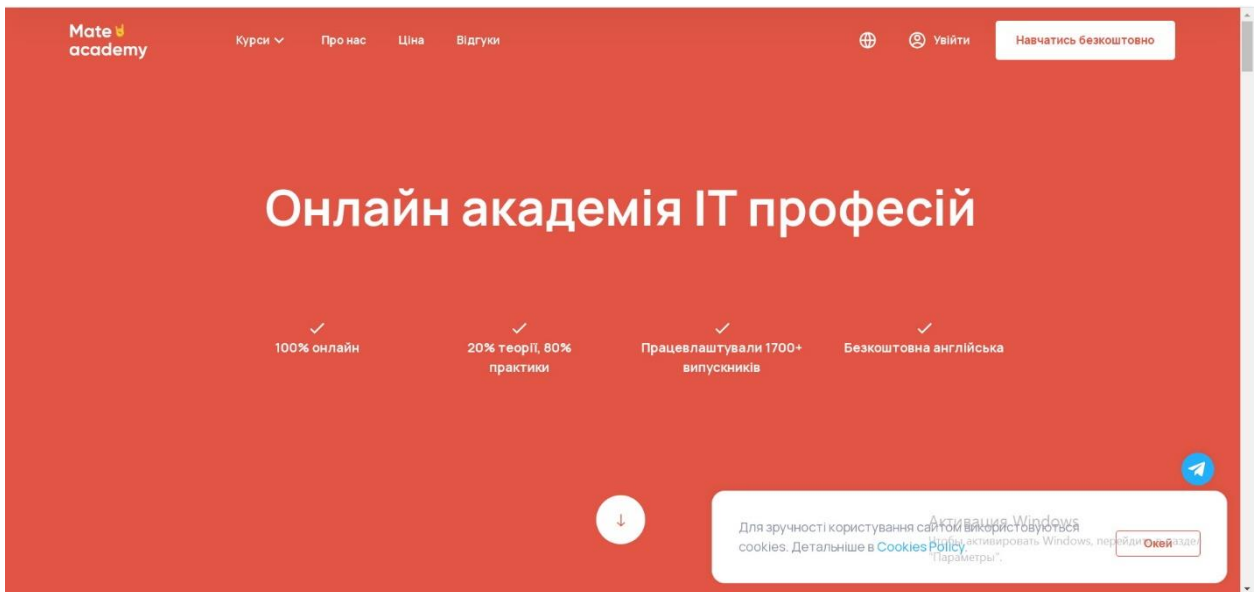


Рис. 1.3. Сайт Mate academy

Сайт Mate academy має приємний інтерфейс і всю необхідну інформацію подалі. Білш класичний мінімалістичний дизайн приємного кольору дає зосередитись на тексті і дає відчуття більшої серйозності (рис .1.3, рис.1.4) [24].

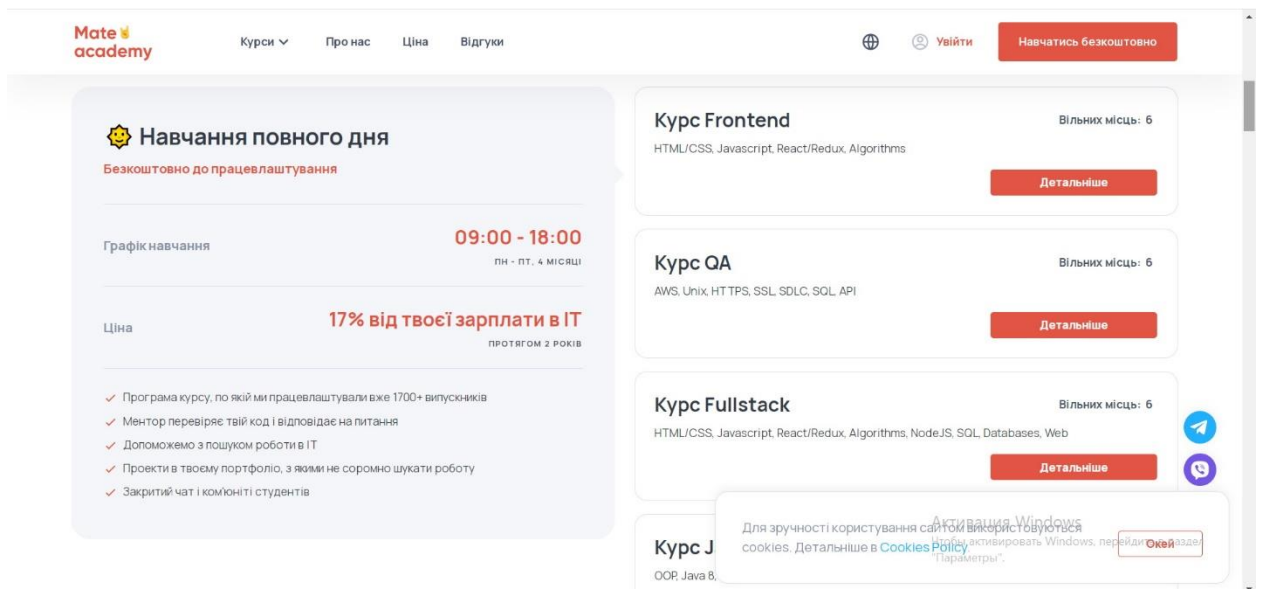


Рис . 1.4. Сайт Mate academy, сторінка курсів

Як бачимо сайти досить схожі, на головній сторінці відразу надають необхідну інформацію заради того, щоб залучити клієнта. Через велику конкуренцію сайти треба розроблювати більш складними, цікавими і привабливими, задля збільшення клієнтів.

1.2. Призначення розробки та галузь застосування

Створення сприятливої умови навчання та надання послуг є дуже важливим завданням. По-перше надання якісного сервісу онлайн освіти збільшить його популярність і внесок у культурний розвиток. По-друге, створення якісного продукту дає можливість бути конкурентно здатним, так як зараз багато фахівців у даній сфері і щоб сервіс був популярним і приносив дохід, він має бути зроблений за допомогою найбільш актуальної інформації.

Тобто нам потрібен привабливий сайт з гарною оптимізацією та адмініструванням для того, щоб адміністратор міг подивитися всі залишені заявки і надати хороший продукт.

Структура Web-сторінок має бути максимально простою, а наповнення – достатньо багатим, щоб відвідувач швидко знайшов потрібну йому інформацію. Крім того, необхідно створювати Web-сторінки так, щоб дизайн не заважав сприйняттю інформації.

Web-сторінки обов'язково слід робити максимально доступними будь-яких пристроях. Web-сторінки повинні швидко завантажуватися та виводитися на екран. Також Web-сторінки не повинні вимагати для відображення жодного додаткового програмного забезпечення.

1.3. Підстава для розробки

Відповідно до освітньої програми, згідно навчального плану та графіків навчального процесу, в кінці навчання студент виконує кваліфікаційну роботу. Тема роботи узгоджується з керівником проекту, випускаючою кафедрою, та затверджується наказом ректора.

Отже, підставами для розробки «виконання кваліфікаційної роботи» є:

- освітня програма 122 «Комп'ютерні науки»;
- навчальний план та графік навчального процесу ;
- наказ ректора Національного технічного університету «Дніпровська політехніка» №268-с від 18.05.2022р.;
- завдання на кваліфікаційну роботу на тему «Розробка веб-орієнтованої інформаційної системи для організації продажу онлайн-курсів з програмування».

1.4. Постановка завдання

Розробка веб-орієнтованої інформаційної системи для організації

продажу онлайн-курсів з програмування. Додаток повинен реалізувати наступні функції:

- легке та доступне адміністрування ;
- мати зручний інтерфейс;
- відображення сторінок для різних девайсів користувача;
- отримання фідбеку.

1.5. Вимоги до програми або програмного виробу

1.5 .1. Вимоги до функціональних характеристик

Для досягнення поставлених цілей програмне забезпечення, що розробляється, повинно підтримувати виконання наступних дій:

- можливість отримати доступ до програми через веб-браузер;
- наявність типової конфігурації, що забезпечує можливість швидкого введення застосунку в експлуатацію;
- програмно-апаратна переносимість;
- надання віддаленого доступу до застосунку на комп'ютері користувача через веб-браузер.

1.5.2 Вимоги до інформаційної безпеки

В ІТ індустрії ніколи не треба забувати про безпеку, так як інтернет небезпечне місце і якщо ігнорувати безпеку, можна легко втратити персональні данні.

Види загроз інформаційної безпеки:

Загроза розкриття інформаційних ресурсів полягає у тому, що дані, інформація і знання стають відомими тим, кому не слід цього знати . У

межах нашої роботи під загрозою розкриття розумітимемо такий стан, коли отриманий несанкціонований доступ до ресурсів системи, при чому йдеться як про відкриті, такі і ті ресурси, які мають обмежений доступ. Ці ресурси мають передаватися один одному і зберігатися у єдиній інформаційній системі.

Загроза порушення цілісності інформаційних ресурсів полягає в умисному антропогенному впливі (модифікація, видалення, зниження) даних, які зберігаються в інформаційній системі суб'єкта управління, а також передаються від даної інформаційної системи до інших.

Загроза збою в роботі самого обладнання може виникнути при блокуванні доступу до одного або декількох ресурсів інформаційної системи. Насправді блокування може бути постійним, так щоб ресурс, що запитується, ніколи не був отриманий, або може викликати затримання в отриманні ресурсу, що запитується, що є достатнім для того, щоб він став некорисним.

Відповідно до викладеного, розглянемо наступні загрози, які загрожують інформаційній безпеці. Розгляд даних загроз робиться з метою продемонструвати, що знання загроз і уразливих місць дозволить організувати адекватну систему управління інформаційною безпекою.

Найбільш частими та небезпечними є ненавмисні помилки користувачів, операторів, системних адміністраторів та інших осіб, які обслуговують інформаційні системи. Іноді такі помилки є загрозами (невірно введені дані, помилка в програмі, котра викликає колапс системи), іноді вони створюють ситуації, якими не лише можуть скористатися зловмисники, а які самі по собі становлять безпосередню небезпеку об'єкта. Яскравим прикладом є уведення невірної інформації в комп'ютер швейцарським оператором на землі, внаслідок чого у небі зіткнулося два

літаки , в одному з яких летіли діти з Росії. Наслідки були трагічними як для пасажирів літака, так і для оператора, якого через деякий час після катастрофи було навмисно вбито .

У цілому ж, за результатами проведених фахівцями з інформаційної безпеки досліджень, понад 65 % шкоди , яка завдається інформаційним ресурсам, є наслідком ненавмисних помилок. Пожежі та землетруси, тобто загрози природного характеру, трапляються набагато рідше. Саме тому, доцільним є акцентування уваги на більшому впровадженні комп'ютерних систем для забезпечення безпеки. Плідною у даному контексті є пропонована програма Національним інститутом стратегічних досліджень "Електронна Україна".

Наступними, за розміром шкоди , можна виділити крадіжки і підлоги, У більшості випадків, суб'єктами вчинення даних дій були штатні працівники цих організацій, які є добре обізнаними у роботі інформаційної системи, а також заходів безпеки.

У цьому аспекті дуже небезпечними є співробітники, які є незадоволеними або не поділяють цінностей тієї організації, де вони працюють. Одним з яскравих прикладів є дія колишнього генерала СБУ, одного з керівників ГУР України Валерія Кравченка, який 18 лютого 2004 року, маючи на руках матеріали з обмеженим доступом, безпідставно надав до них доступ іншим особам, зокрема журналістам Дойче Веле.

Для уникнення некоректної роботи програми необхідно реалізувати:

- семантичний та синтаксичний контроль вхідних даних;
- обробку виняткових ситуацій;
- виведення повідомлень про помилки;
- можливість повторного введення даних;
- можливість безперервної роботи протягом не менше 120 годин (5 діб);

- платформну незалежність.

1.5.3 Вимоги до складу та параметрів технічних засобів

Для нормального функціонування даного застосунку необхідно, щоб персональний комп'ютер, на якому, буде функціонувати система інтрнет додатка, відповідала наступним вимогам:

- процесор класу Intel Pantium з тактовою частотою не менш 2.4 ГГц та двома ядрами;
- доступ до мережі Internet;
- не менше 8 GB оперативної пам'яті;
- 500 GB вільного місця на жорсткому диску;
- клавіатура;
- маніпулятор "миша".

Наведені вище технічні характеристики є рекомендованими, тобто при наявності інших технічних засобів, що будуть не нижче зазначених, розроблений застосунок буде функціонувати. Тож якщо характеристики будуть такими, то можна не перейматись, щодо вимог надійності, швидкості обробки даних і безпеки, висунутими замовником.

1.5.4 Вимоги до інформаційної та програмної сумісності

Для нормального функціонування програми необхідно, щоб програмне забезпечення персонального комп'ютера, на якому буде функціонувати веб-орієнтована система, відповідало наступним вимогам:

- операційна система Windows (7+), Linux, MacOS;
- веб-браузер Firefox / Google Chrome / Opera / Microsoft Edge / Safari.

РОЗДІЛ 2

ПРОЕКТУВАННЯ ТА РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 . Функціональне призначення системи

За завданням кваліфікаційної роботи було реалізовано веб-орієнтовану інформаційну систему для організації продажу онлайн-курсів з програмування.

Призначення розробленої системи:

- забезпечити просте адміністрування сайту;
- надати можливість обмінюватися даними та запитами між користувачем та адміністратором;
- забезпечити просте користування сайтом;
- надати можливість користувачам знайти цікавий для них курс і дати можливість вкладати в свій розвиток і майбутнє.

Розроблена програма реалізує наступні функції :

- контактування з адміністратором сайту;
- оформлення замовлень.

Для досягнення поставленої задачі розроблене програмне забезпечення підтримує виконання таких операцій:

- надання віддаленого доступу до застосунку через веб-браузер на комп'ютері або іншому девайсі користувача;
- зчитування вхідних даних з хмарних сервісів та бази даних;
- формування замовлення та відправка на адміністративну частину;
- забезпечення авторизації користувача для перегляду контенту.

2.2. Опис застосованих математичних методів

Оскільки особливості предметної галузі розв'язуваної задачі не передбачають застосування математичних методів, при розробці системи відображення та управління контенту інтернет-магазину математичні методи не використовувалися.

2.3. Опис використаних технологій та мов програмування

В даній кваліфікаційній роботі було використані наступні технології та мови програмування:

- HTML;
- CSS;
- Python;
- Client Server Architecture;
- Figma.

Розглянемо більш детально технології та мови, що представлені в списку .

HTML

Ця мова є базовою в галузі технологій створення сайтів, оскільки відносно легка в освоєнні. Але надмірна простота є його недоліком . HTML (від англійської Hyper Text Markup Language – мова розмітки гіпертексту) чудово відповідав вимогам раннього періоду розвитку технологій створення сайтів, але з подальшим його розвитком виникли суттєві проблеми. HTML надає такі можливості:

- Видавати мережеві документи із заголовками, текстом, таблицями,

- списками, фотографіями тощо;
- Отримувати інформацію з Мережі через гіпертексту посилання при натисканні кнопки;
 - Створювати форми посилки запитів на віддалені комп'ютери, щоб здійснювати пошук інформації, здійснювати бронювання, замовляти товари тощо;
 - Включати електронні таблиці , відео кліпи, аудіо кліпи, та інші програмні програми безпосередньо до їхніх документів.

Історія розробки HTML досить тривала. У кожній його версії розробники намагалися добитися того, щоб HTML-сторінки читалися усіма браузерами, на всіх комп'ютерних платформах .

Мова HTML призначений для формування та оформлення сторінок тексту, який призначений для перегляду в Інтернеті за допомогою спеціальних програм - браузерів (основні з них Netscape Navigator (фірми Netscape) та Internet Explorer (фірми Microsoft)).

В даному випадку , під сторінкою мається на увазі окремий файл зі спеціальними мітками, які вказують браузеру, як саме повинен бути відображений на екрані та в роздруківці текст, який міститься в цьому файлі. Крім цього, можуть бути ще посилання на файли з графічними зображеннями, які будуть відображатися разом з текстом під час перегляду .

Для формування файлу HTML немає необхідності використання потужного текстового процесора, зручніше використання, наприклад, редактора Notepad (Блокнот). Вміст HTML-сторінок представлений у гіпертекстовому вигляді, що передбачає наявність у документі , що відображається на екрані, виділених певним чином місць, клацнувши мишею по яких можна переміститися в іншу частину цієї ж сторінки або на іншу сторінку.

CSS

Каскадні таблиці стилів (CSS) - спеціальна мова, що використовується для відображення сторінок, написаних мовами розмітки даних. Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів. HTTP-сервер - відкритий веб-сервер Інтернет для UNIX - подібних, Microsoft Windows, Novell NetWare та інших операційних систем. Apache розроблюється та підтримується спільнотою розробників відкритого програмного забезпечення під керівництвом Apache Software Foundation. В 1996 році Apache обійшов NCSA HTTPd із того часу є найбільш популярним веб-сервером у світі. Web-сервер Apache є самостійним, некомерційним, вільно розповсюджуваним продуктом. Продукт підтримує безліч можливостей, багато з яких реалізовані як скомпільовані модулі, які розширюють основні функціональні можливості. Вони різняться від серверної підтримки мов програмування до схем аутентифікації. Існують інтерфейси для підтримки мов програмування Perl, Python, Tcl і PHP. Популярні методи стискування на Apache включають зовнішній модуль mod_gzip, створений для зменшення розміру веб-сторінок, переданих по HTTP. Функції віртуального хостингу дозволяють одній інсталяції Apache обслуговувати різні веб-сайти. Наприклад, одна машина, з однією інсталяцією Apache може одночасно містити www.example.com, www.test.com, server.test.com і так далі. Apache передусім використовується для передачі через HTTP статичних та динамічних веб-сторінок у всесвітній павутині. Багато веб-застосунків спроектовано, зважаючи на середовище і можливості, які надає цей веб-сервер. Продукт може працювати в якості кешувального проксі-сервера, що дозволяє істотно підвищити продуктивність роботи користувачів локальної мережі при роботі з документами, розташованими в Інтернет. Можна задавати такі параметри і налаштування проксі-сервера: типи файлів, які необхідно кешувати або навпаки, не включати в кеш, максимальний обсяг дискового простору, відведений під кеш, періодичний перегляд і індексування бази даних кеша з

метою вивільнення дискового простору шляхом видалення застарілих об'єктів.

Python

Python проста у використанні, та водночас повноцінна мова програмування, що надає набагато більше засобів для структурування і підтримки великих програм, ніж shell. З іншого боку, вона краще за C обробляє помилки, і, будучи мовою дуже високого рівня, має вбудовані типи даних високого рівня, такі як гнучкі масиви і словники, ефективна реалізація яких на C потребує значних витрат часу.

Завдяки більш загальним типам даних, Python застосовують до більш широкого кола задач, ніж Awk і навіть Perl, у той ж час багато речей на мові Python робляться настільки ж просто.

Python дозволяє розбивати програми на модулі, що потім можуть бути використані в інших програмах. Python поставляється з великою бібліотекою стандартних модулів, які можна використовувати як основу для нових програм або як приклади при вивченні мови. Стандартні модулі надають засоби для роботи з файлами, системними викликами, мережними з'єднаннями і навіть інтерфейсами до різних графічних бібліотек.

Python - інтерпретована мова, що дозволяє заощадити значну кількість часу, що зазвичай витрачається на компіляцію. Інтерпретатор можна використовувати інтерактивно, що дозволяє експериментувати з можливостями мови, писати шаблони програм або тестувати функції при розробці "знизу-вверх". Він також зручний як настільний калькулятор. Python дозволяє писати дуже компактні й зручні для читання програми. Програми, написані мовою Python, звичайно значно коротші еквівалента на C або C++ з декількох причин:

- типи даних високого рівня дозволяють Вам виразити складні операції однією інструкцією;

- групування інструкцій виконується за допомогою відступів замість фігурних дужок;
- немає необхідності в оголошенні змінних.

Python розширювана мова: знання C дозволяє додавати нові функції, що вбудовуються, або модулі для виконання критичних операцій з максимальною швидкістю або написання інтерфейсу до комерційних бібліотек, доступним тільки у двійковій формі. Інтерпретатор мови Python може бути вбудований у програму, написану на C, і використовувати його як розширення або командну мову для цієї програми. Python використовується в даний час десятками тисяч програмістів в усьому світі, і число людей, що використовують його, швидко зростає, подвоюється і потроюється щороку. Python приваблює користувачів з ряду причин. Він використовується для розробки програм і дозволяє провести розробку набагато швидше, ніж традиційні мови типу C, C++ або Java. Ця мова працює однаково добре на Windows, UNIX, Macintosh, і OS/2, може використовуватися, для легкої розробки як малих додатків чи сценаріїв, так і для розгортання великих програм. Python пропонує доступ до могутнього і легкого у використанні комплекту 29 інструментальних засобів графічного інтерфейсу користувача. Традиційні машинні мови типу C і Pascal мають ряд характеристик, наприклад, сувору типізацію, базові типи, складні (і звичайно довгі) цикли, і потреба у великих кількостях кодів для виконання відносно малих задач. Java досить новий, але розділяє більшість характеристик, включених у цей перелік. Програмісти, знайомі з традиційними мовами погодяться, що відсутність суворої типізації полегшує роботу з Python.

Client Server Architecture

В даному веб-ресурсі використана архітектура «клієнт-сервер».

Архітектура клієнт-сервер є одним із архітектурних шаблонів

програмного забезпечення та є домінуючою концепцією у створенні розподілених мережних застосунків і передбачає взаємодію та обмін даними між ними. Вона передбачає такі основні компоненти:

— набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;

— набір клієнтів, які використовують сервіси, що надаються серверами;

— мережа, яка забезпечує взаємодію між клієнтами та серверами.

Сервери є незалежними один від одного. Клієнти також функціонують паралельно і незалежно один від одного. Немає жорсткої прив'язки клієнтів до серверів. Більш ніж типовою є ситуація, коли один сервер одночасно обробляє запити від різних клієнтів; з іншого боку, клієнт може звертатися то до одного сервера, то до іншого. Клієнти мають знати про доступні сервери, але можуть не мати жодного уявлення про існування інших клієнтів (рис. 2.1).

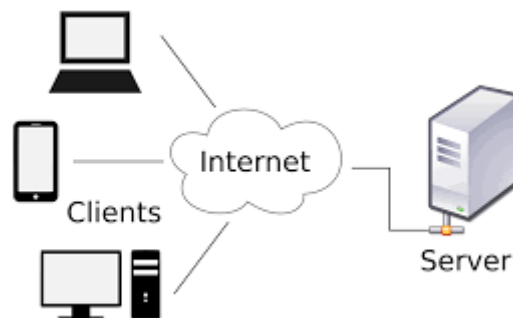


Рис.2.1 Архітектура «клієнт-сервер»

Дуже важливо ясно уявляти, хто або що розглядається як «клієнт». Можна говорити про клієнтський комп'ютер, з якого відбувається звернення до інших комп'ютерів. Можна говорити про клієнтське та серверне програмне забезпечення. Нарешті, можна говорити про людей, які бажають за допомогою відповідного програмного та апаратного забезпечення отримати доступ до тієї чи іншої інформації.

Загальноприйнятим є положення, що клієнти та сервери - це перш за все програмні модулі . Найчастіше вони знаходяться на різних комп'ютерах, але бувають ситуації, коли обидві програми - і клієнтська , і серверна, фізично розміщуються на одній машині ; в такій ситуації сервер часто називається локальним .

Модель клієнт-серверної взаємодії визначається перш за все розподілом обов'язків між клієнтом та сервером . Логічно можна відокремити три рівні операцій:

— рівень представлення даних, який по суті являє собою інтерфейс користувача і відповідає за представлення даних користувачеві і введення від нього керуючих команд ;

— прикладний рівень, який реалізує основну логіку застосунку і на якому здійснюється необхідна обробка інформації ;

— рівень управління даними, який забезпечує зберігання даних та доступ до них. Дворівнева клієнт-серверна архітектура передбачає взаємодію двох програмних модулів - клієнтського та серверного. В залежності від того, як між ними розподіляються наведені вище функції, розрізняють :

— модель тонкого клієнта, в рамках якої вся логіка застосунку та управління даними зосереджена на сервері. Клієнтська програма забезпечує тільки функції рівня представлення;

— модель товстого клієнта, в якій сервер тільки керує даними, а обробка інформації та інтерфейс користувача зосереджені на стороні клієнта. Товстими клієнтами часто також називають пристрої з обмеженою потужністю: кишенькові комп'ютери, мобільні телефони та ін .

Трирівнева клієнт-серверна архітектура, яка почала розвиватися з середини 90-х років, передбачає відділення прикладного рівня від

управління даними. Відокремлюється окремий програмний рівень, на якому зосереджується прикладна логіка застосунку. Програми проміжного рівня можуть функціювати під управлінням спеціальних серверів застосунків, але запуск таких програм може здійснюватися і під управлінням звичайного вебсервера. Нарешті, управління даними здійснюється сервером даних .

Для роботи з системою користувач використовує стандартне програмне забезпечення - звичайний браузер. Це позбавляє його необхідності завантажувати та інсталювати спеціальні програми. Але користувачеві слід надати в розпорядженні інтерфейс, який дозволяв би йому взаємодіяти з системою і формувати запити до неї. Форми, що визначають цей інтерфейс, розміщуються на веб-сторінках та завантажуються разом з ними.

Веб-оглядач формує запит та пересилає його до сервера, який здійснює обробку. При необхідності сервер викликає серверні програмні модулі, які забезпечують обробку запиту і в разі потреби звертаються до сервера даних . Сервер даних здійснює операції з даними, що зберігаються в системі та складають її інформаційну основу. Зокрема, він може здійснити вибірку з інформаційної бази відповідно до запиту та передати її модулю проміжного рівня для подальшої обробки . Дані, з якими працює сервер даних, найчастіше організовані як реляційна база даних.

Найчастіше веб -сервер і серверні модулі проміжного рівня розміщуються на одному комп'ютері, хоч і являють собою окремі і логічно незалежні програмні модулі.

Figma

Figma це графічний редактор для веб-дизайну . У Figma можна створювати:

– інтерактивні прототипи сайтів і мобільних додатків;

– елементи інтерфейсу - іконки, кнопки, меню, вікна, форми
зворотного зв'язку;

– векторні ілюстрації .

У Figma всі документи зберігаються в хмарі . Завдяки цьому в редакторі можна колективно працювати над макетами і відкривати їх за посиланням , без скачування.

Можна заходити в Figma через браузер або ж завантажити програму на комп'ютер . Вона підходить як для Windows, так і для Mac. У десктопній версії можна працювати офлайн, а коли з'являється доступ до Інтернету , зміни синхронізуються.

Вихідні документи зберігаються в хмарі , тому не потрібно пересилати макети, викладати їх в хмару і контролювати версії. Просто заходиш в акаунт Figma і бачиш оригінал. Якщо співробітника немає на роботі, не доведеться шукати макет на його комп'ютері - все в командному доступі.

Одна з головних привелегій Figma як для веб-розробки – формування коду CSS для кожного окремого блоку та можливість його повного застосування у поєднанні з HTML.

Як і в Google Docs, в Figma можна працювати над документом разом: давати доступ на перегляд і редагування, паралельно працювати над макетом - на екрані буде видно курсори різного кольору.

Завдяки цьому всі учасники проекту краще розуміють контекст . Наприклад, дизайнер розробляє прототип додатка, а UX копірайтер прямо в Figma пише тексти для інтерфейсу.

Інструмент безкоштовний для індивідуальних користувачів.

Якщо ви працюєте в акаунті самі, всі функції безкоштовні. Просто реєструєтесь і починаєте працювати. Можна давати доступ на перегляд

документа - це теж безкоштовно.

Figma дозволяє зробити клікабельно версію сайту або мобільного додатку, створюючи безліч екранів. Можна протестувати і налаштувати будь-яку функцію: кнопки, спливаючі меню, анімовані і модальні вікна.

Всі екрани розміщені на одній сторінці. Перемикається між вікнами не потрібно. Ви створюєте зв'язку і бачите, як робота з інтерфейсом виглядає в реальності.

Редактор створений в першу чергу для веб-дизайну. Тому в ньому незручно працювати з поліграфією. Щоб підготувати макет для банера, листівки або ролл-апу для друку, потрібно створити в Figma файл, завантажити в іншу програму, що призначена для поліграфії. І тільки потім його віддавати на друк.

Ще одне обмеження Figma в тому, що макети з Photoshop можна імпортувати прямо. Один з варіантів - перевести їх в формат Sketch, а потім перенести в Figma.

2.4 Опис структури системи та алгоритмів її функціонування

Чітка та логічна структура – це перше, що потрібно врахувати, перш ніж робота над сайтом почне готуватися. Правильна структура веб-сайту позитивно впливає на рейтинг пошукової системи та сприйняття користувачами веб-сайту. Веб-сайти – це, по суті, інформаційні ресурси, опубліковані для використання людьми. Подібно до книг, вони можуть бути незначними або великими за обсягом. Але на відміну від книг, вміст веб-сайту може бути структурований найрізноманітнішими способами, використовуючи свободу вираження поглядів та дизайну. У той же час веб-сайти повинні слідувати найкращим галузевим практикам UX та звичкам та

очікуванням користувачів, тому свобода досить обмежена. Структура веб-сайту є ключовою для того, яку інформацію використовує цільова аудиторія. Тому його доводиться будувати за великою кількістю правил. Структура веб-сайту стосується того, як веб-сайт побудований, як всі окремі підсторінки на ньому пов'язані між собою та яка ієрархія дотримується. Хороша структура зробить пошук інформації інтуїтивним та простим у використанні. Коли створюється структура веб-сайту, потрібно думати про те, як відвідувачі використовуватимуть веб-сайт і орієнтуються в його вмісті. Фокус зміщений з того, що публікувати, на те, як і чому. Структура веб-сайту є основним поняттям з трьох основних точок зору: цілі власника бізнесу, досвід відвідувачів сайту, який отримує від цього вигоду, та пошукові системи, які сканують, класифікують та обслуговують вміст потрібним користувачам. 29 Коли інформація на веб-сайті добре структурована і підтримується відповідним дизайном, усі задоволені – власник сайту, користувачі та пошукові системи.

Хороша структура веб-сайту:

- Підтримує цілі власника бізнесу і підвищує очікувані результати. Потрібно поставити чіткі цілі щодо його ефективності. Ці цілі мають бути інтегровані та виражені структурою сайту;
- Підвищує досвід користувача. Структура веб-сайту тісно впливає на досвід роботи веб-сайту. Коли інформація структурована в логічний спосіб і все інтуїтивно доступно, користувачі безперешкодно пересуваються через вміст, щоб знайти та використовувати те, що вони шукають;
- SEO оптимізація.

Пошукові системи (Google) працюють з інструментами, які сканують вміст сайту та сортують його для користувачів. Веб-сайти, зручні для сканування, мають гарну структуру, і тому їх рейтинг вищий.

Без чіткого бачення того, як спланувати веб-сайт та побудувати його структуру, неможливо виконати веб-дизайн та розробку.

Структуру веб-сайту можна визначити як структурну проекцію інформаційного простору, що забезпечує інтуїтивний доступ до вмісту. Зокрема, структура веб-сайту є важливою при розробці системного підходу до процесу розробки належної навігації.

Роль розробника полягає у створенні веб-сайту, який має якісний інтерфейс і простий у використанні. Якісна структура веб-сайту покращує зручність використання веб-сайту, полегшуючи користувачам пошук того, що вони шукають. Структура веб-сайту впорядковує вміст на своєму сайті (домашня сторінка, категорії, окрема сторінка, публікації в блозі).

Проектування внутрішньої структури сайту – це процес дослідження, розробки та вдосконалення всіх аспектів взаємодії користувачів із продуктом компанії для задоволення користувачів. Мета полягає в тому, щоб покращити 30 взаємодію користувача з товаром за допомогою результатів тестування, в кінцевому підсумку створити продукт, який є корисним і цінним, а також легким у отриманні, приємним у використанні.

UX – це не про стилістичні рішення, кольору, шрифти і т.п. Більшою мірою це про архітектуру сайту, його структуру, логічну зв'язку в різних блоках. Саме це формує так званий «користувальницький досвід» (звідси і абревіатура UX, від «User eXperience»).

Дизайн UX сприяє створенню продукту, який забезпечує ефективну взаємодію з користувачами. Він включає різні дисципліни, такі як візуальний дизайн, дизайн взаємодії, зручність використання тощо.

UX повинен вирішувати такі завдання:

- простий пошук всієї необхідної користувачеві інформації;
- інтуїтивне орієнтування в структурі сайту як такого;

- відпрацювання типових сценаріїв поведінки відвідувачів на сайті;
- легкість здійснення користувачем цільових для бізнесу дій. Виходячи з таких цілей і проводиться розробка UX-дизайну сайту.

UI – це візуальний вигляд товару, розробники вирішують, як візуально оформити продукт для ефективного користувацького досвіду. Хороший дизайн інтерфейсу допомагає залучити користувачів та забезпечує приємний досвід роботи з продуктом. Елементи дизайну інтерфейсу включають такий вміст, як документи, текст, зображення, відео; форми, включаючи кнопки, теги, текстові поля, прапорці, випадаючі списки, графічний дизайн; та поведінку, подібна до того, що станеться, коли користувач натискає / перетягує / вводить. Це вимагає від розробника відчуття естетики та технічних навичок. Творча роль, мета дизайнера інтерфейсу – створити переконливий, красивий користувацький інтерфейс, який викликає емоційний відгук користувача.

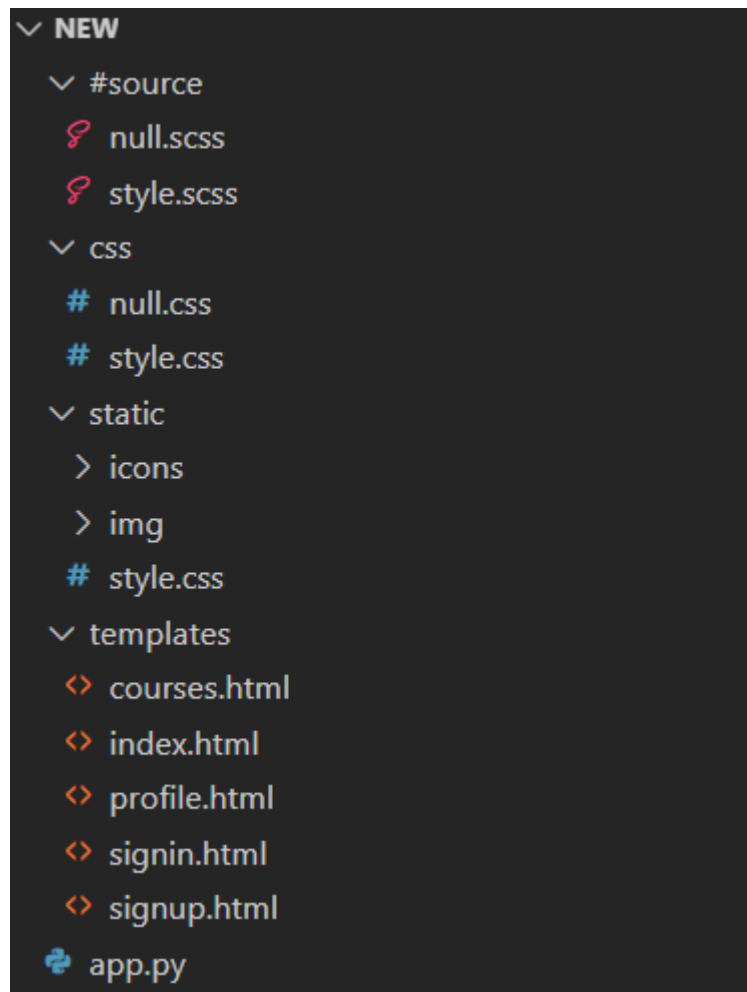


Рис. 2.2. Структура файлової системи

Структура папок даного сайту представлена на рис. 2.2. Вона включає в себе папки:

- source – папка стилів scss;
- Css – папка стилів css;
- Static – папка в якій знаходяться всі зображення і іконки, а також файл style.css який відповідаю за стиль цих зображень;
- Templates – папка де знаходяться всі сторінки сайту.

Сайт включає такі сторінки:

- Courses.html – сторінка вибору кірсів;
- Index.html – головна сторінка;
- Profile.html – сторінка профіля користувача;

- Signin.html – сторінка реєстрації ;
- Signup.html – сторінка входу в профіль користувача .

Файл app.py знаходиться в основній папці new і відповідає за форму обратного зв'язку , оптимізацію сторінок між собою і коректну роботу сайту.

2.5 Обґрунтування та організація вхідних та вихідних даних програми

Вхідною інформацією в даній системі є дані, що вносяться користувачем за допомогою клавіатури та миші під час вибору вкладок сайту, інформація, що ідентифікує користувача під час авторизації: ім'я, номер телефона, електронна пошта, а також форма обратного зв'язку. До вхідної інформації належать дані, що вносить адміністратор про курси: опис, фото, відео.

Вихідними даними є інформація, що надається користувачеві під час відображення даних з сайту та інформація що відображається під час замовлення курсу.

2.6 . Опис роботи розробленої системи

2.6.1 . Використані технічні засоби

Рекомендована конфігурація, що забезпечує цілодобову роботу програми з резервуванням даних:

- процесор класу AMD Ryzen 5 4600 3.8 (4.2)GHz 16MB;
- материнська плата MSI PRO B660-A DDR4;
- модулі пам'яті Crucial CT8G4DFRA266 DDR4 - 8ГБ 2666, DIMM, Ret;

- система охолодження 92mm Direct Air Flow;
- SSD диск Kingston SSDNow A400 240 GB (SA400S37/240G);
- рідкокристалічний монітор з діагоналлю не менше 17 ";
- доступ до мережі Internet;
- маніпулятор "миша";
- клавіатура .

Наведені вище технічні характеристики є рекомендованими, тобто при наявності технічних засобів не нижче зазначених , розроблений програмний виріб буде функціонувати відповідно до вимог щодо надійності, швидкості обробки даних і безпеки, висунутими замовником.

2.6.2. Використані програмні засоби

Даний проект реалізований на таких мовах як HTML, CSS, Python.

Для створення сайтів є багато програм і засобів для реалізації цього. Це редактори html, серверів та скриптів, а також редактори каскадних таблиць css. Я користувався Microsoft Visual Studio Code.

Visual Studio Code (VS Code) – редактор вихідного коду , розроблений Microsoft для Windows, Linux та macOS. Позиціонується як «легкий» редактор коду для кросплатформної розробки веб- та хмарних програм. Включає в себе відладчик, інструменти для роботи з Git, підсвічування синтаксису, IntelliSense та засоби для рефакторингу . Має широкі можливості для кастомізації: теми користувача , поєднання клавіш і файли конфігурації. Розповсюджується безкоштовно , розробляється як програмне забезпечення з відкритим вихідним кодом, але готові зборки розповсюджуються під ліцензією пропріетарної.

Для клієнта необхідними програмними та технічними засобами є

мобільний телефон, ПК або інший девайс з підключенням до Інтернету та браузером

2.6.3. Виклик та завантаження програми

Для користування сайтом потрібно завантажити його на хостинг та придбати доменне ім'я. Для використання сайту локально потрібно відкрити файл app.ru, запустити його і перейти за посиланням у браузер.

2.6.4. Опис інтерфейсу користувача

При завантаженні сайту в браузері відкривається головна сторінка на якій, згідно з зовнішньою структурою розташовані елементи. Шапка сайту містить логотип, головне меню, профіль користувача (рис. 2.2).

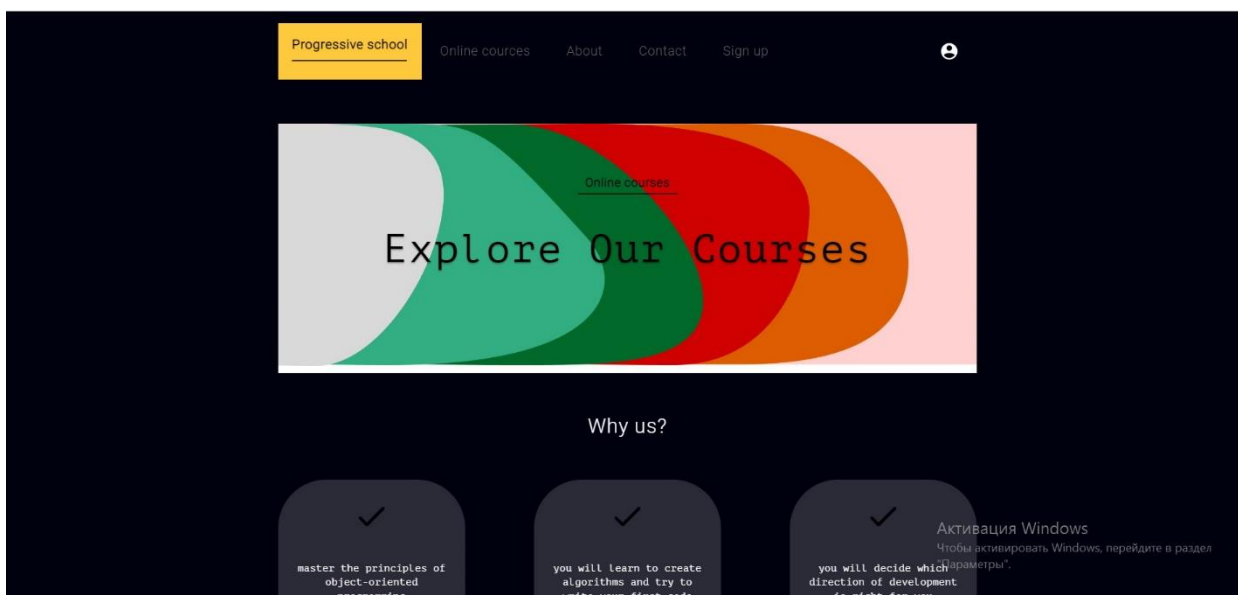


Рис. 2.3. Шапка сайту

Головне меню складається з:

- клікабельного логотипу, який при натисканні повертає до головної сторінки сайту (рис.2.3);
- кнопки «Online courses», яка здійснює перехід до сторінки вибору курсів (рис.2.4);

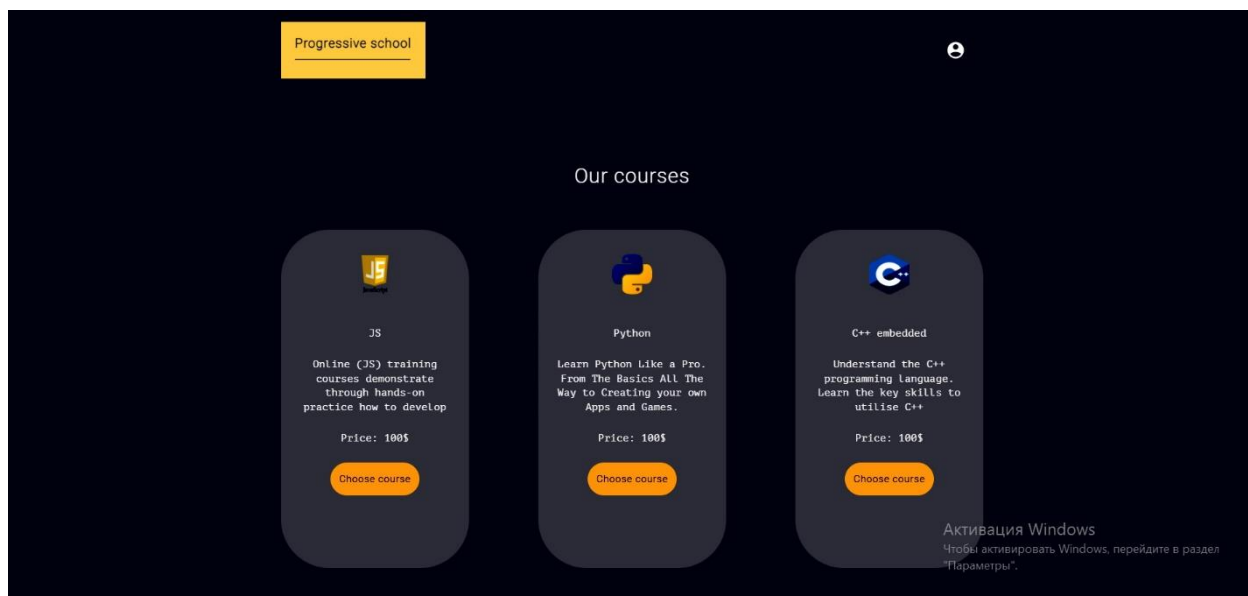


Рис. 2.4. Сторінка курсів.

- кнопки «About», яка здійснює перехід нижче по сайту до презентації курсу (рис.2.5);

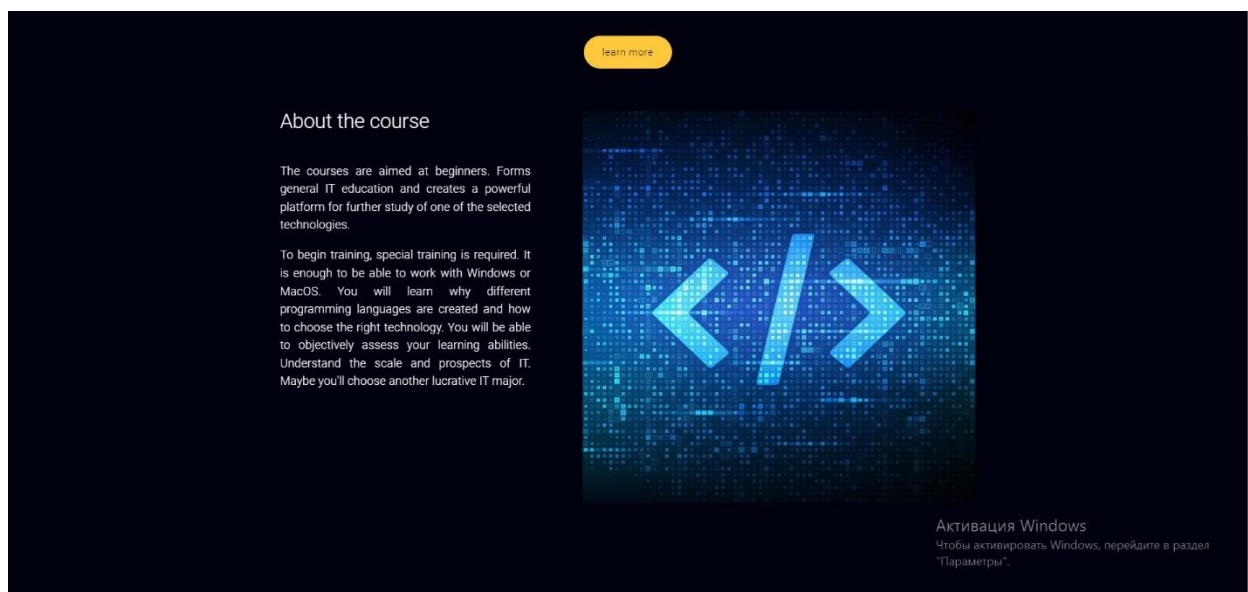


Рис. 2.5. Презентація курсу.

- кнопка «Contact», яка здійснює перехід до низу сайту, де розміщена форма зворотнього зв'язку і посилання до соціальних мереж (рис.2.6);

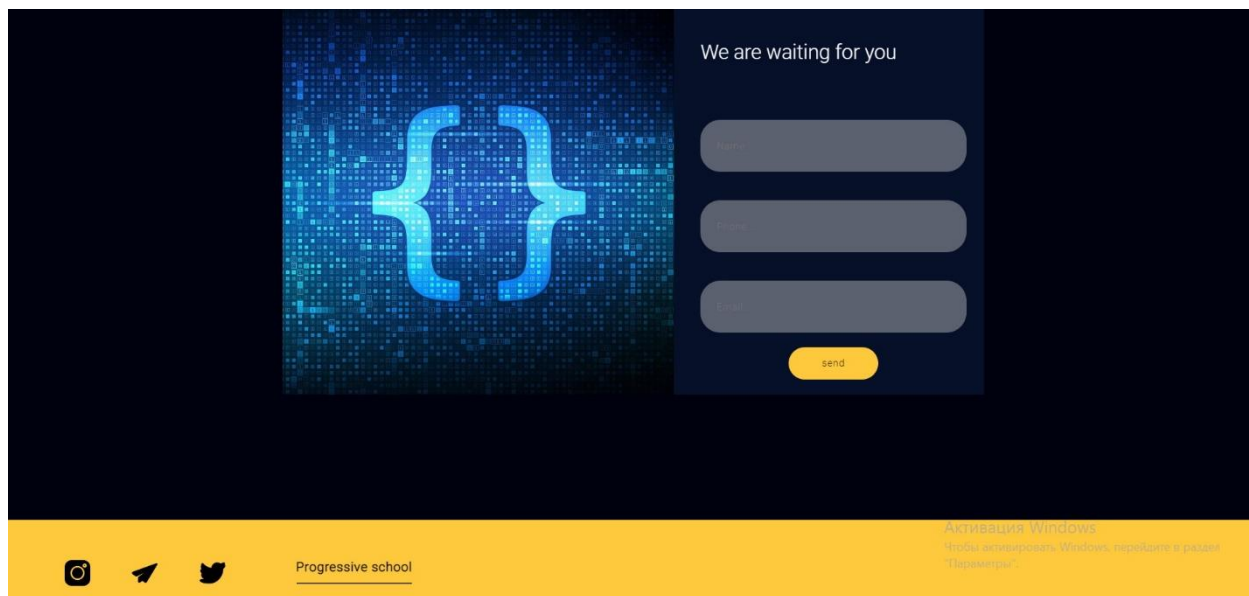


Рис. 2.6. Низ сайту.

- кнопка «Sign in» яка здійснює перехід до сторінки реєстрації (рис.2.7). А також кнопка змінюється на «Log out», після того як користувач увійшов в профіль і при натисканні дає можливість вийти зі свого профілю (рис. 2.10);

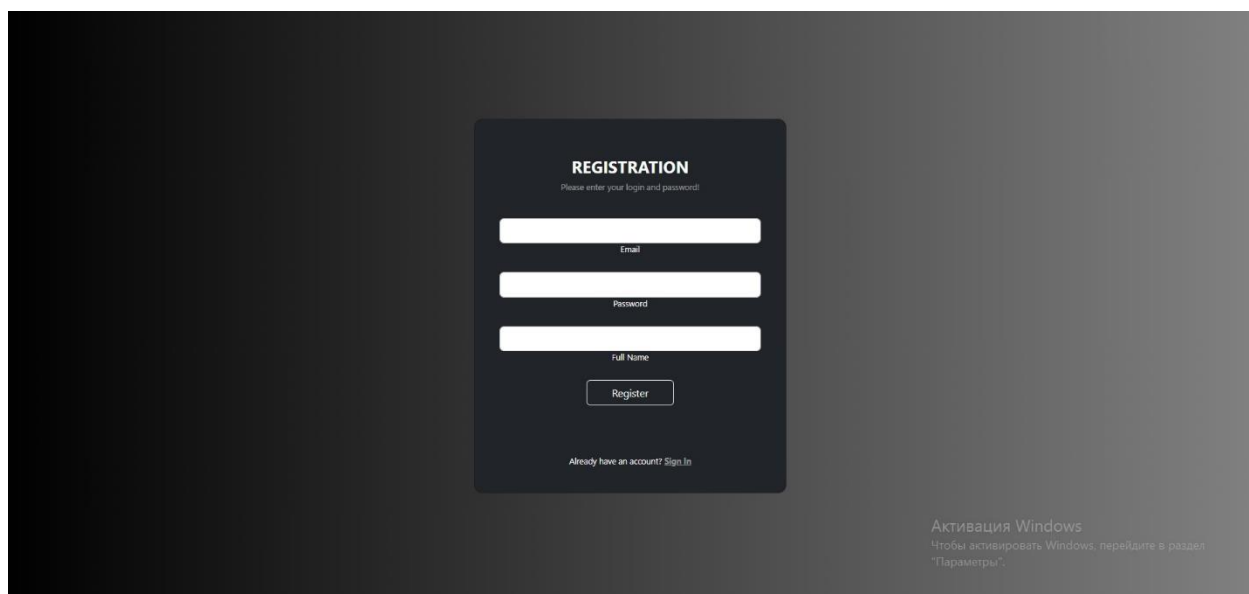


Рис. 2.7. Сторінка реєстрації.

- кнопка профілю користувача яка має два вигляди при натисканні. До реєстрації і після (рис.2.8, рис.2.9).

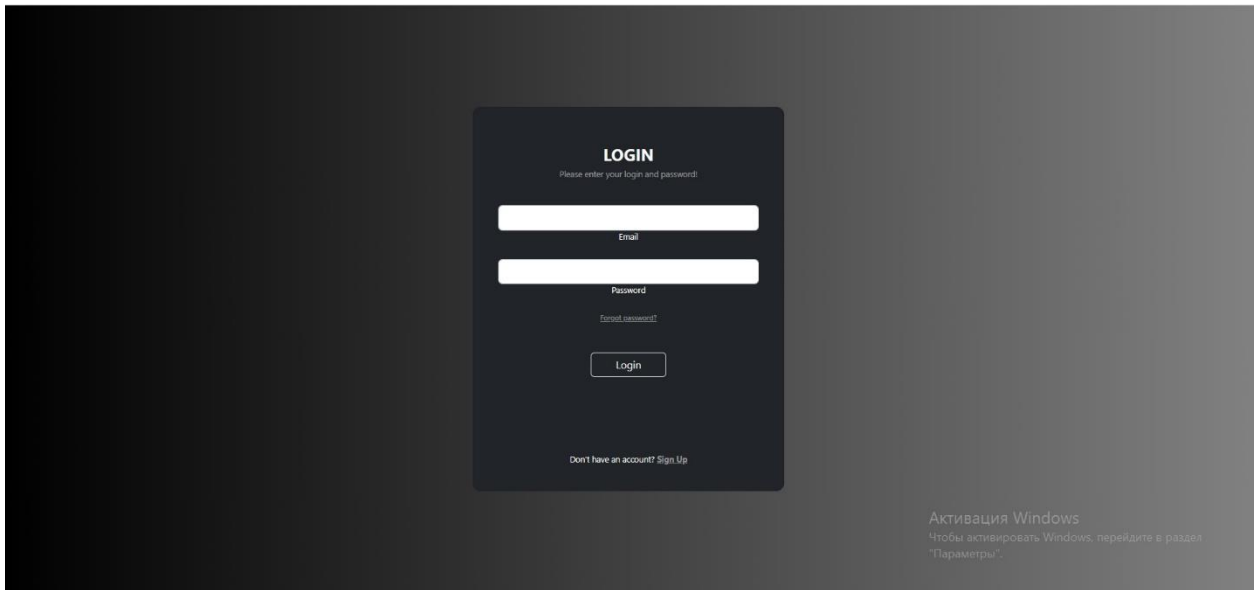


Рис. 2.8. Сторінка входу в профіль

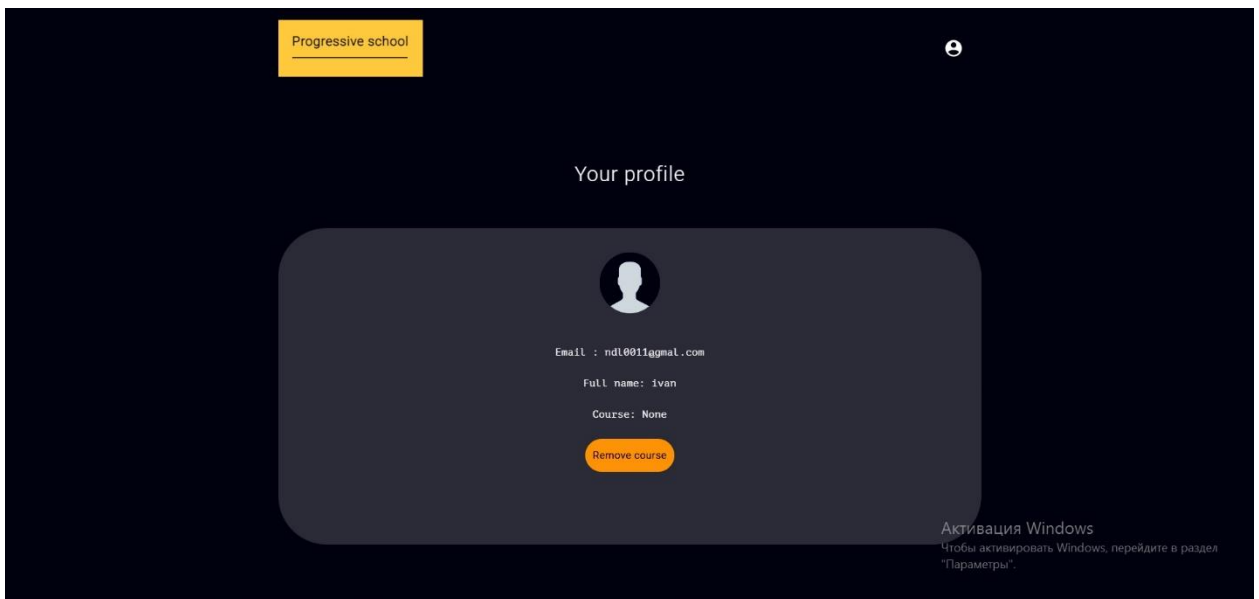


Рис. 2.9. Профіль користувача

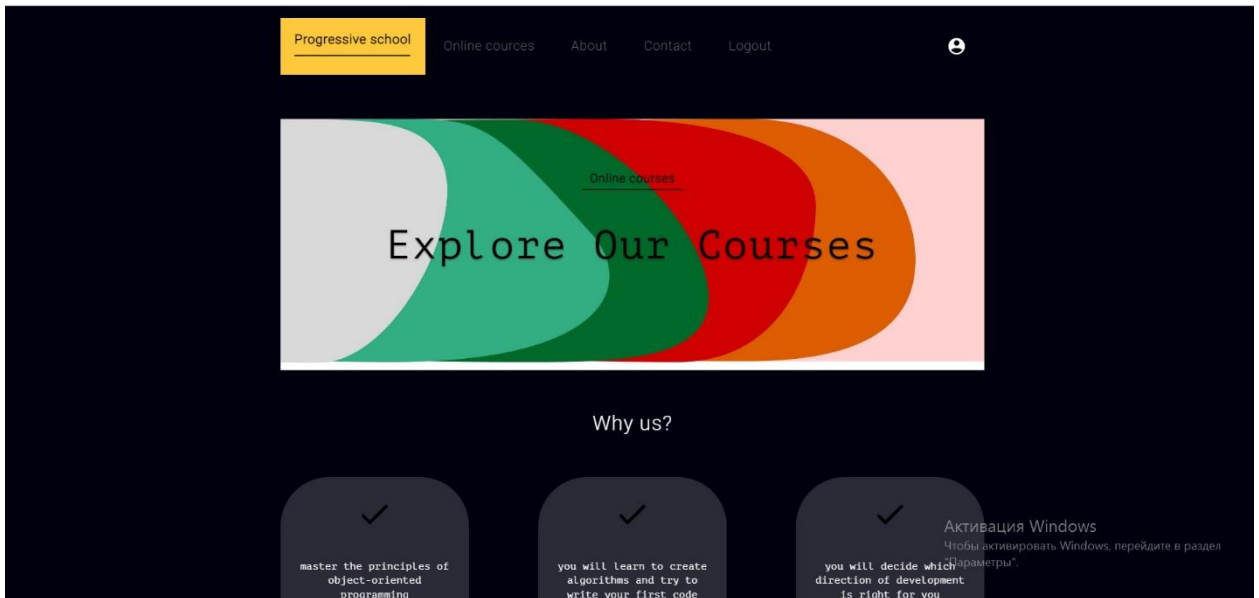


Рис. 2.10. Шапка сайту після входу в профіль

Якщо опуститися нижче головної сторінки ми побачимо початок презентації курсу, де означені переваги обрати самі ці курси (рис.2.11).

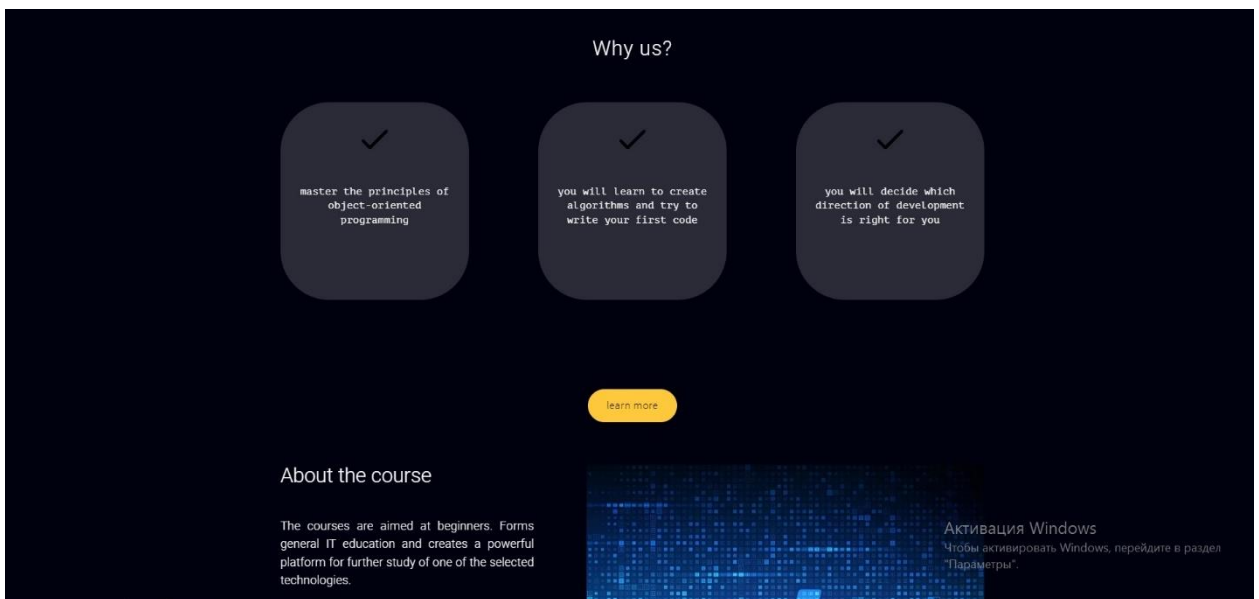


Рис. 2.11. Переваги онлайн школи.

Далі продовжується презентація курсу, де розповідається детальніше про сам курс (рис.2.4). Також бачимо кнопку «learn more» при натисканні на яку нас перебрисує на сторінку вибору курсів (рис.2.3).

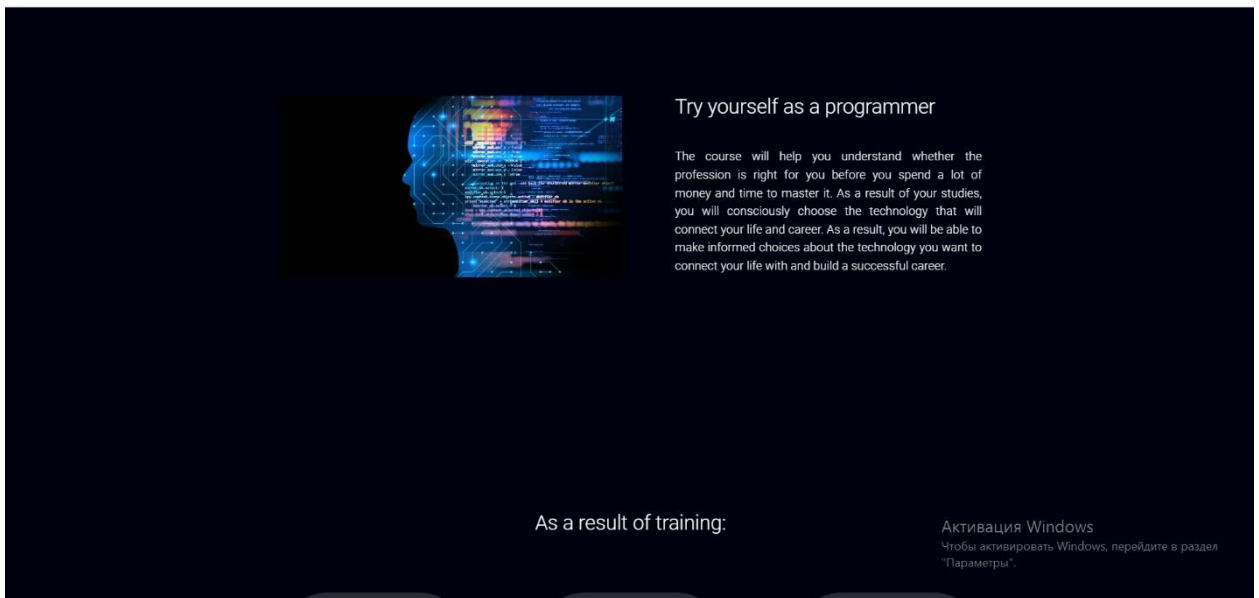


Рис.2.11. Презентація спеціальності програмист.

Далі продовжується презентація де користувачу пропонують спробувати себе в програмуванні (рис.2.11).

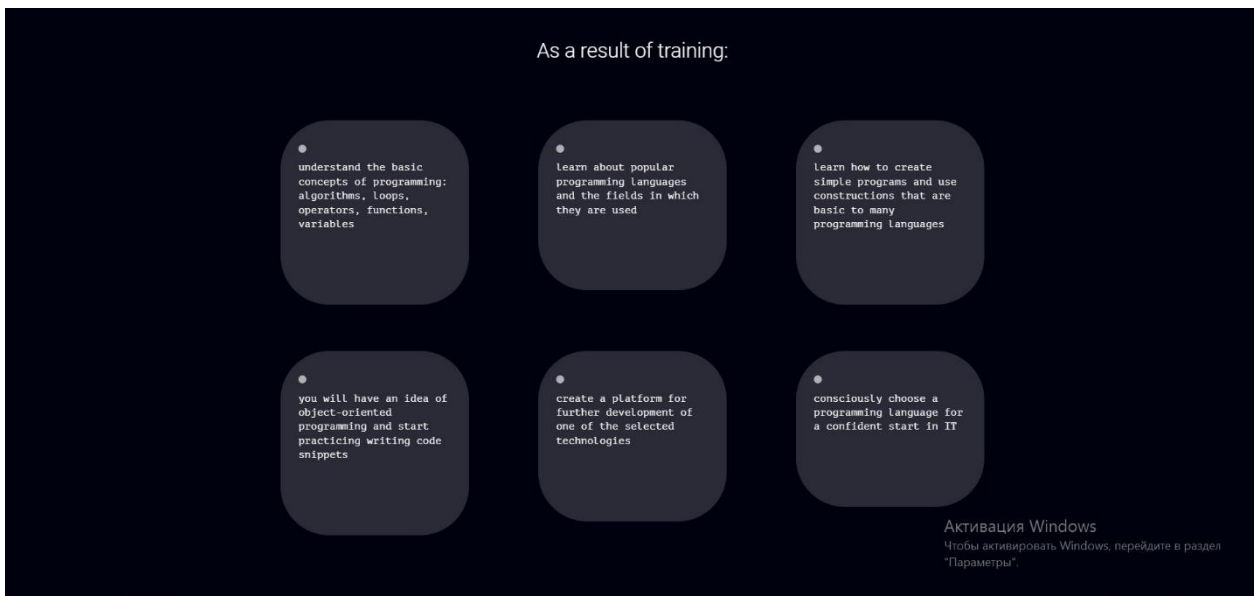


Рис.2.12. Презентація результатів навчання.

Опускаючись нижче ми бачимо результати яких можна досягнути після проходження курсу (рис.2.12).

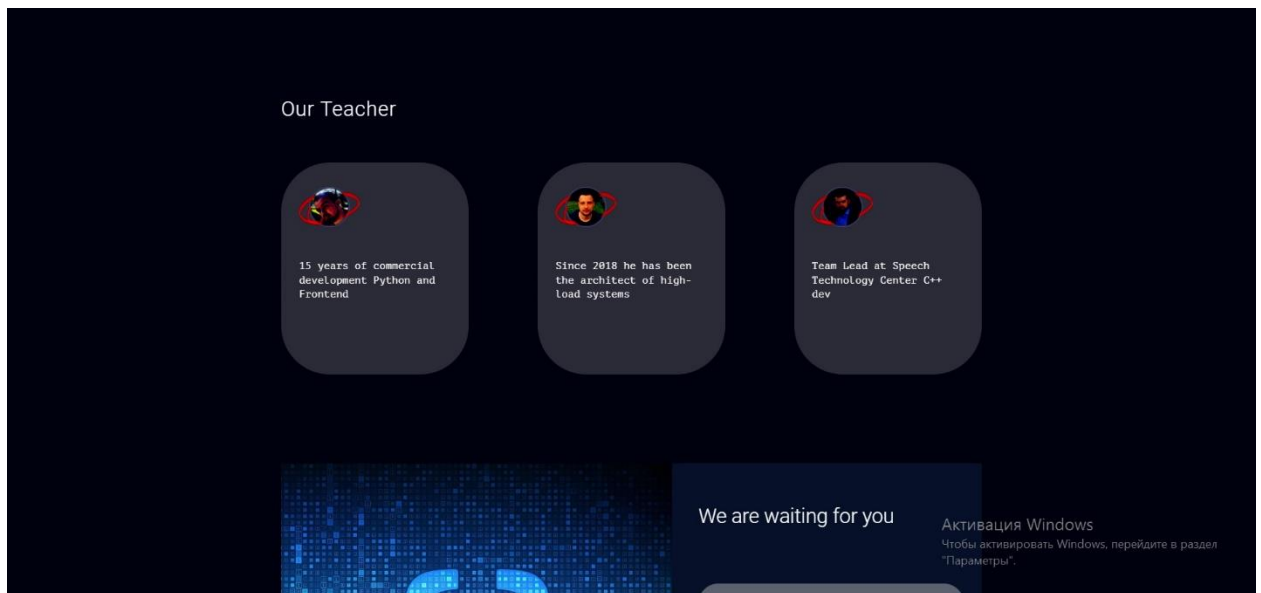


Рис. 2.13. Презентація вчителів курсу.

Нижче ми бачимо презентацію вчителів цього курсу (рис.2.13).

На цьому презентація курсу завершується і ми спускаємось до низу сайту де користувачу дають можливість залишити свої данні, якщо його зацікавить пропозиція і перейти за посиланням у соціальні мережи (рис.2.5).

При розробці додатку було вирішено не використовувати БД через складність підключення до серверу, складність розробки адміністративної панелі і складність оптимізації, а також знаходження простого і локанічного рішення, яке нічим не поступається в реалізації необхідних задач. Тому я віддав перевагу методу відсилення вхідних даних на поштову скриньку за реалізованому на мові Python.

Форма зворотного зв'язку відсилається на вказану поштову скриньку (рис.2.14).

REMINDER:Callback  Входящие x



andreysdoruk@yahoo.com

кому: мне ▾

 английский ▾ > русский ▾ [Перевести сообщение](#)

Hello, Andrii! Just wanted to callback

Email: mediandrey@gmail.com

Phone: 389456454643

 Ответить  Переслать

Рис. 2.14. Форма зворотного зв'язку

РОЗДІЛ 3

ЕКОНОМІЧНИЙ РОЗДІЛ

3.1 Розрахунок трудомісткості та вартості розробки програмного продукту

Початкові дані:

1. передбачуване число операторів програми – 2420;
2. коефіцієнт корекції програми в ході її розробки – 0,05;
3. коефіцієнт складності програми – 1,25;
4. годинна заробітна плата програміста– 135 грн/год;
5. коефіцієнт збільшення витрат праці в наслідок недостатнього опису задачі. 1,4;
6. коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності – 1,1;
7. вартість машино-годин ЕОМ – 18 грн/год.

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:

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

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, яке розробляється.

Умовне число операторів (підпрограм):

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

де q - передбачуване число операторів (2420);

C - коефіцієнт складності програми (1,25);

p - коефіцієнт корекції програми в ході її розробки (0,05).

Звідси умовне число операторів в програмі:

$$Q = 1,25 \cdot 2420 \cdot (1 + 0,05) = 3176,25$$

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

$$t_u = \frac{Q \cdot B}{75 \cdot 85 \cdot k}, \text{ людино-годин, } (3.3)$$

де B - коефіцієнт збільшення витрат праці внаслідок недостатнього опису задачі;

k - коефіцієнт кваліфікації програміста, обумовлений від стажу роботи з даної спеціальності. При стажі роботи від 3 до 5 років він складає

1,1.

Прийmemo збільшення витрат праці внаслідок недостатнього опису завдання не більше 50% ($B = 1.4$). З урахуванням коефіцієнта кваліфікації $k = 1,1$, отримуємо витрати праці на вивчення опису завдання:

$$t_u = (3176,25 \cdot 1,4) / (75 \cdot 1,1) = 53,9 \text{ людино-годин}$$

Витрати праці на розробку алгоритму рішення задачі визначаються за формулою:

$$t_a = \frac{Q}{(20\dots25) \cdot k}, \text{ людино-годин, (3.4)}$$

де Q – умовне число операторів програми;

k – коефіцієнт кваліфікації програміста.

Підставивши відповідні значення в формулу (3.4), отримаємо:

$$t_a = \frac{3176,25}{22 \cdot 1,1} = 131,25, \text{ людино-годин.}$$

Витрати на складання програми по готовій блок-схемі:

$$t_n = \frac{Q}{(20\dots25) \cdot k}, \text{ людино-годин, 3.5}$$

$$t_n = \frac{3176,25}{23 \cdot 1,1} = 125,54 \text{ людино-годин.}$$

Витрати праці на налагодження програми на ЕОМ:

– за умови автономного налагодження одного завдання:

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

$$t_{\text{отл}} = \frac{3176,25}{4*1.1} = 721,87 \text{ людино-годин.}$$

– за умови комплексного налагодження завдання:

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

$$t_{\text{отл}}^k = 1,5 * 721,87 = 1082,8, \text{ людино-годин.}$$

Витрати праці на підготовку документації визначається за формулою:

$$t_{\text{д}} = t_{\text{др}} + t_{\text{до}}, \text{ людино-годин, (3.8)}$$

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

$$t_{\text{др}} = \frac{Q}{(15...20)*k}, \text{ людино-годин, (3.9)}$$

$t_{\text{до}}$ – трудомісткість редагування, печатки й оформлення документації:

$$t_{\text{до}} = 0,75 * t_{\text{др}}, \text{ людино-годин, (3.10)}$$

Підставляючи відповідні значення, отримуємо:

$$t_{\text{др}} = \frac{3176,25}{17*1.1} = 169,85, \text{ людино-годин.}$$

$$t_{\text{до}} = 0,75 * 169,85 = 127,38, \text{ людино-годин.}$$

$$t_d = 169,85 + 127,38 = 297,23, \text{ людино-годин.}$$

Повертаючись до формули (3.1), отримаємо повну оцінку трудомісткості розробки програмного забезпечення:

$$t = 50 + 53,9 + 131,25 + 125,54 + 721,87 + 297,23 = 1379,79, \text{ людино-годин.}$$

3.2 Розрахунок витрат на створення програми

Витрати на створення ПЗ $K_{\text{ПО}}$ включають витрати на заробітну плату виконавці програми $Z_{\text{ЗП}}$ і витрат машинного часу, необхідного на налагодження програми на ЕОМ:

$$K_{\text{ПО}} = Z_{\text{ЗП}} + Z_{\text{МВ}}, \text{ грн. (3.11)}$$

Заробітна плата виконавців визначається за формулою:

$$Z_{\text{ЗП}} = t * C_{\text{Пр}}, \text{ грн. (3.12)}$$

де t – загальна трудомісткість, людино-годин.

$C_{\text{Пр}}$ – середня годинна заробітна плата програміста, грн/година.

З урахуванням того, що середня годинна зарплата програміста становить 80 грн/год, отримуємо:

$$З_{зп} = 1379,79 * 135 = 186\,271,65 \text{ грн}$$

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

$$З_{мв} = t_{отл} * C_{мч} , \text{ грн. (3.14)}$$

де $t_{отл}$ – трудомісткість налагодження програми на ЕОМ, год;

$C_{мч}$ – вартість машино-годин ЕОМ, грн/год (18 грн/год).

Підставивши в формулу (3.3) відповідні значення, визначимо вартість необхідного для налагодження машинного часу:

$$З_{мв} = 721,87 * 18 = 12\,993,66 \text{ грн.}$$

Звідси витрати на створення програмного продукту:

$$K_{по} = 186\,271,65 + 12\,993,66 = 199\,265,31 \text{ грн.}$$

Очікуваний період створення ПЗ:

$$T = \frac{t}{B_k * F_p}, \text{ міс. (3.15)}$$

де B_k – число виконавців (дорівнює 1);

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

Звідси витрати на створення програмного продукту:

$$T = \frac{1379,79}{1 \cdot 176} \approx 7,83 \text{ міс.}$$

Висновок: програмне забезпечення розроблено для забезпечення доступу користувачів до основних програм та пропозицій компанії продажу онлайн курсів, ефективної взаємодії між потенційними споживачами та компанією, створення позитивного іміджу компанії, підвищення продажу курсів. Вартість даного програмного забезпечення близько 199 265,31 тис. грн і не вимагає додаткових витрат як при розробці програми. Очікуваний час розробки становить 1379,79 годин, тобто 7,83 місяці. Цей термін пов'язаний зі значним числом операторів, і включає час на дослідження і розробку алгоритму вирішення поставленого завдання, програмування по готовому алгоритму, налагодження програми і підготовку документації.

ВИСНОВКИ

Метою кваліфікаційної роботи була розробка веб-орієнтованої інформаційної системи для організації продажу онлайн-курсів з програмування. Розробка такої системи є дуже актуальною в цей час, тому що зараз треба переносити усе в онлайн простір, щоб у людей була можливість вчитись будь де і щоб не які зовні фактори не заважали розвитку.

Дана система створена розроблена за допомогою каскадних таблиць стилів (CSS), мови розмітки HTML та мови програмування Python. В результаті веб-ресурс має детальну інформацію про курси, ціни, послуги, що надаються, інформацію про вчителів та форма замовлення послуги.

В роботі було досліджено та проаналізовано найбільш важливу і актуальну інформацію щодо розробки веб-ресурсів, а саме основні принципи створення вебресурсів, їх структуру і функціональність, взаємодію основних компонентів. Також були розглянуті новітні та найбільш перспективні веб-технології, які з успіхом вже використовуються користувачами по всьому світі.

В процесі розробки веб-ресурсу та оформлення роботи на практиці були закріплені теоретичні знання, вдосконалені навички програмування, проведено оформлення технічної документації в текстових та графічних редакторах .

Розроблений сайт є інформативним, зручним в користуванні з яскравим та зрозумілим інтерфейсом.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Удовик І.М., Коротенко Л.М., Шевцова О.С. Методичні вказівки з виконання економічного розділу в дипломних проектах студентів спеціальності “Програмне забезпечення”. – М : НТУ «Дніпровська політехніка», 2013. 16 с.
2. Євдокимов Н. В. Основи тематичної оптимізації. - М .: Вільямс, 2007. 30с
3. Кантор И. Браузер:документ, события, интерфейсы. –М. : Эксмо, 2019. 309 с
4. Персіваль Г. Python. Розробка на основі тестування. - М .: ДМК, 2018. 622с
5. Дейт К. Дж. Введення в системи баз даних. 8-е изд. - М .: Вільямс, 2005. 1328с.
6. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов. / В.Дронов. – СПб.: БХВ-Петербург, 2011. – 416 с.
7. HTML и CSS. Путь к совершенству. / Б. Хеник – СПб.: Питер, 2011. – 336 с.
8. Матвієнко, О.В. Internet-технології: проектування Webсторінки: Навчальний посібник для студентів вузів / О. В Матвієнко, І. Л Бородкіна. - 2-е вид., перероб. и доп. . - К.: ЦНЛ, 2004. - 154 с.
9. Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Р. Никсон. - Москва: Машиностроение, 2016. - 688 с.
10. Пасічник О.Г. Основи веб-дизайну. - К.:ВНУ. - 2009. – 354 с.
11. Проценко О.Б. Web-програмування та web-дизайн. Технологія XML: Навчальний посібник. – Суми: Видавництво СумДУ, 2009. – 166 с.
12. Холл М., Браун Л. Программирование для Web.- М.: Издательский дом «Вильямс», 2002.- 1259 с.

13. Эд Титтел, Джефф Ноубл. HTML, XHTML и CSS для чайников, 7-е издание. HTML, XHTML & CSS For Dummies, 7th Edition. — М.: «Диалектика», 2011. — 400 с.
14. Фримен, Элизабет Изучаем HTML, XHTML и CSS / Элизабет Фримен , Эрик Фримен. - М.: Питер, 2016. - 720 с.
15. Скотт Хокинс. Администрирование веб-сервера Apache и руководство по электронной коммерции = Apache Web Server Administration and e-Commerce Handbook / Скотт Хокинс. – Москва: Вильямс, 2001. – 336 с. – (ISBN 0-13-089873-2).
16. Методичні рекомендації щодо написання, оформлення та представлення учнівських науково-дослідницьких робіт учнів – членів Малої академії наук України / Г.Г. Півняк, Л.М. Коротенко, І.М. Удовик, Є.М. Головня – Д.: ДВНЗ «Національний гірничий університет», 2017. – 24 с.
17. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення : ДСТУ 3008-95. – Чинний від 1996–01–01. – К. : Держстандарт України, 1996. – 39 с.
18. Бусигін Б.С., Коротенко Г.М., Коротенко Л.М. Прикладна інформатика. Підручник для студентів комп'ютерних спеціальностей. – Дніпропетровськ: Видавництво НГУ, 2004. – 559 с.
19. Бусыгин Б.С., Дивизинюк М.М., Коротенко Г.М., Коротенко Л.М. Введение в современную информатику. Учебник. – Севастополь: Издательство СТУЭИП, 2005. – 644 с. [Электронный ресурс] - Режим доступа: <https://learn.javascript.ru/screencast/react> Дата звернення: 18.06.2022.
20. Database Concepts / D.Kroenke, D. Auer, S. Vandenberg, R. Yoder., 2019. – 552 с
21. Farrington T. UX Design 2020: The Ultimate Beginner's Guide to User Experience / T. Farrington, T. Brooks, L. Ferrante., 2020.
22. Duckett J. HTML and CSS : Design and Build Websites / Jon Duckett.

23. Сайт онлайн школи степ. [Електронний ресурс] - Режим доступу:
<https://itstep.dp.ua/Rozrobka-programnogo-zabezpesenna> Дата звернення:
18.06.2022.
24. Сайт онлайн школи степ. [Електронний ресурс] - Режим доступу:
https://mate.academy/?utm_source=google&utm_medium=cpc&utm_term=%D0%BA%D1%83%D1%80%D1%81%D0%B8%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F&utm_content=587882340630&utm_campaign=gs_generic_ua_mxd&gclid=CjwKCAjw77WVBhBuEiwAJ-YoJI9t06JQ_Aw53YwU0y2URkB_yeEcoZuO_dboskopstr4YX3cNY47TMBoCDC8QAvD_BwE Дата звернення: 18.06.2022.

КОД ПРОГРАМИ

Index.html // ГОЛОВНИЙ HTML файл

```

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

  <head>
    <meta charset="UTF-8">
    <title>Progressive school</title>
    <link rel="stylesheet" href="{{url_for('static', filename = 'style.css')}}">
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=PT+Mono&display=swap"
rel="stylesheet">
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link
href="https://fonts.googleapis.com/css2?family=PT+Mono&family=Roboto:wght@300&display
=swap" rel="stylesheet">
    <meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0, user-scalable=0">
  </head>
  <body>
    <div class="wrapper">
      <header>
        <div class="container">
          <div class="header">
            <div class="logo">
              <a href="/"><div
class="logo__text">Progressive school</div></a>
              <div class="logo__vector">
                
              </div>
            </div>
          </div>
        </div>
      </header>
    </div>
  </body>
</html>

```

```

</div>
<div class="menu">
    <ul class="menu__list">
        <a href="/courses"><li>Online courses</li></a>
        <a href="#yech"><li>About</li></a>
        <a href="#foo"><li>Contact</li></a>
        {% if not session['logged_in'] %}
        <a href="/register"><li>Sign up</li></a>
        {% endif %}
        {% if session['logged_in'] %}
        <a href="/logout"><li>Logout</li></a>
        {% endif %}
    </ul>
</div>
<div class="other">
    <ul class="other__list">
        <li>
            <a href="/login"></a>
        </li>
    </ul>
    <div class="form-popup" id="myForm">
        <form action="#" class="form-
container">
            <h2>We are waiting for
you</h2>
            <input type="text"
placeholder="Name" name="name" required>

```



```

src="{{url_for('static', filename = 'icons/tick.png')}}" alt="">
</div>
<div
class="columns__text">master the principles of object-oriented programming</div>
</div>
</div>
<div class="columns__element">
<div
class="columns__content">
<div
class="columns__img">

</div>
<div
class="columns__text">you will learn to create algorithms and try to write your first code</div>
</div>
</div>
<div class="columns__element">
<div
class="columns__content">
<div
class="columns__img">

</div>
<div
class="columns__text">you will decide which direction of development is right for you</div>
</div>
</div>
</div>
<div class="button">
<a href="/courses">
<button
class="open_button" >learn more</button>
id="yech"

```

```

        </a>
    </div>
</div>
<div class="about">
    <div class="about__row">
        <div class="about__element">
            <div class="about__content">
                <div
class="about__main">About the course</div>
                <div class="about__text">The
courses are aimed at beginners. Forms general IT education and creates a powerful platform for
further study of one of the selected technologies.</div>
                <div
class="about__text__span">To begin training, special training is required. It is enough to be able
to work with Windows or MacOS. You will learn why different programming languages are
created and how to choose the right technology. You will be able to objectively assess your
learning abilities. Understand the scale and prospects of IT. Maybe you'll choose another lucrative
IT major.</div>
            </div>
        </div>
    </div>
    <div class="about__element">
        <div class="about__content">
            
        </div>
    </div>
</div>
<div class="about__row">
    <div class="about__element">
        <div class="about__content">
            
        </div>
    </div>
    <div class="about__element">
        <div class="about__content">

```

```

class="about__main">Try yourself as a programmer</div>
</div>
<div class="about__text">The
course will help you understand whether the profession is right for you before you spend a lot of
money and time to master it. As a result of your studies, you will consciously choose the
technology that will connect your life and career.
As a result, you will be
able to make informed choices about the technology you want to connect your life with and build
a successful career.</div>
</div>
</div>
</div>
</div>
<div class="result">
<div class="result__main">As a result of
training:</div>
<div class="columns__row">
<div class="columns__element">
<div class="columns__content">
<div
class="columns__vector"></div>
<div
class="columns__text">understand the basic concepts of programming: algorithms, loops,
operators, functions, variables</div>
</div>
</div>
<div class="columns__element">
<div class="columns__content">
<div
class="columns__vector"></div>
<div
class="columns__text">learn about popular programming languages and the fields in which they
are used</div>
</div>
</div>
</div class="columns__element">

```

```

class="columns__vector"></div>
class="columns__text">learn how to create simple programs and use constructions that are basic
to many programming languages</div>
class="columns__vector"></div>
class="columns__text">you will have an idea of object-oriented programming and start practicing
writing code snippets</div>
class="columns__vector"></div>
class="columns__text">create a platform for further development of one of the selected
technologies</div>
class="columns__vector"></div>
class="columns__text">consciously choose a programming language for a confident start in
IT</div>

```

```

        </div>
    </div>
    <div class="why__text">Our Teacher</div>
    <div class="why__block">
        <div class="columns__row">
            <div class="columns__element">
                <div class="columns__content">
                    <div class="columns__img">
                        
                    </div>
                    <div
class="columns__text">15 years of commercial development Python and Frontend</div>
                </div>
            </div>
            <div class="columns__element">
                <div class="columns__content">
                    <div class="columns__img">
                        
                    </div>
                    <div
class="columns__text">Since 2018 he has been the architect of high-load systems</div>
                </div>
            </div>
            <div class="columns__element">
                <div class="columns__content">
                    <div class="columns__img">
                        
                    </div>
                    <div
class="columns__text">Since 2018 he has been the architect of high-load systems</div>
                </div>
            </div>
        </div>
    </div>

```

```

        </div>
        <div
class="columns__text">Team Lead at Speech Technology Center C++ dev</div>
        </div>
        </div>
        </div>
        </div>
        </div>
        <form action="/index" method="POST">
        <div class="form">
            <div class="form__row">
                <div class="form__element">
                    <div class="form__content">
                        
                    </div>
                </div>
                <div class="form__element">
                    <div class="form__content">
                        <div id="foo"
class="form__title">We are waiting for you</div>
                        <ul>
                            <li>
                                <input
type="text" placeholder="Name..." name="name" required>
                            </li>
                            <li>
                                <input
type="text" placeholder="Phone..." name="phone" required>
                            </li>
                            <li>
                                <input
type="text" placeholder="Email..." name="email" required>
                            </li>
                        </ul>
                    </div>
                </div>
            </div>
        </div>
    </div>

```



```

<div class="container">
  <div class="header">
    <div class="logo">
      <a href="/"><div
class="logo__text">Progressive school</div></a>
      <div class="logo__vector">
        
      </div>
    </div>
    <div class="other">
      <ul class="other__list">
        <li>
          <a class="none_l1"
href="/login"></a>
        </li>
      </ul>
      <div class="form-popup" id="myForm">
        <form action="#" class="form-
container">
          <h2>We are waiting for
you</h2>
          <input type="text"
placeholder="Name" name="name" required>
          <input type="tel"
placeholder="Phone" name="telephone" required>
          <input type="mail"
placeholder="Email" name="mail" required>
          <button type="submit"
class="btn" onclick="closeForm()">send</button>
        </form>
      </div>
    </div>
  </div>
</div>

```

```

        </div>
    </header>
    <div class="content">
        <div class="container">
            <div class="why">
                <div class="why__text">Our courses</div>
                <div class="why__block">
                    <div class="columns__row">
                        <div class="columns__element">
                            <div class="columns__content">
                                <div class="columns__img">
                                    
                                </div>
                                <div
class="columns__text">JS</div>
                                <br><br><br>
                                <div  class="columns__text">Online
(JS) training courses demonstrate through hands-on practice how to develop</div>
                                <br><br><br>
                                <div  class="columns__text">Price:
100$</div>
                                <br><br><br>
                                <div class="button">
                                    <a href="/js">
                                        <button  id="corses1"
class="open_button" >Choose course</button>
                                    </a>
                                </div>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
    <div class="columns__element">
        <div class="columns__content">
            <div class="columns__img">

```

```

width="100" src="https://upload.wikimedia.org/wikipedia/commons/thumb/c/c3/Python-logo-notext.svg/2048px-Python-logo-notext.svg.png" alt="">
</div>
<div
class="columns__text">Python</div>
<br><br><br>
<div class="columns__text">Learn
Python Like a Pro. From The Basics All The Way to Creating your own Apps and Games.</div>
<br><br><br>
<div class="columns__text">Price:
100$</div>
<br><br><br>
<div class="button">
<a href="/python">
<button id="corses1"
class="open_button" >Choose course</button>
</a>
</div>
</div>
</div>
<div class="columns__element">
<div class="columns__content">
<div class="columns__img">

</div>
<div class="columns__text">C++
embedded</div>
<br><br><br>
<div
class="columns__text">Understand the C++ programming language. Learn the key skills to utilise
C++</div>
<br><br><br>

```

100\$</div>

<div class="columns__text">Price:

<div class="button">

<button id="corses1"

class="open_button" >Choose course</button>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

<footer>

<div class="footer__row">

<div class="footer__element">

<div class="footer__content">

<div class="footer__icon" >

</div>

</div>

</div>

<div class="footer__element">


```

        <link
href="https://fonts.googleapis.com/css2?family=PT+Mono&family=Roboto:wght@300&display
=swap" rel="stylesheet">
        <meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0, user-scalable=0">
    </head>
    <body>
        <div class="wrapper">
            <header>
                <div class="container">
                    <div class="header">
                        <div class="logo">
                            <a href="/"><div
class="logo__text">Progressive school</div></a>
                            <div class="logo__vector">
                                
                            </div>
                        </div>
                    </div>
                    <div class="other">
                        <ul class="other__list">
                            <li>
                                <a class="none_1"
href="/login"></a>
                            </li>
                        </ul>
                    </div>
                    <div class="form-popup" id="myForm">
                        <form action="#" class="form-
container">
                            <h2>We are waiting for
you</h2>
                            <input type="text"
placeholder="Name" name="name" required>

```



```

        <meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0, user-scalable=0">
    </head>
    <body>
<section class="vh-100 gradient-custom">
    <div class="container py-5 h-100">
        <div class="row d-flex justify-content-center align-items-center h-100">
            <div class="col-12 col-md-8 col-lg-6 col-xl-5">
                <div class="card bg-dark text-white" style="border-radius: 1rem;">
                    <div class="card-body p-5 text-center">

                        <div class="mb-md-5 mt-md-4 pb-5">

                            <h2 class="fw-bold mb-2 text-uppercase">Login</h2>
                            <form action="/login" method="POST">
                                <p class="text-white-50 mb-5">Please enter your login and password!</p>

                                <div class="form-outline form-white mb-4">
                                    <input type="email" name="username" class="form-control form-control-lg" />
                                    <label class="form-label" for="typeEmailX">Email</label>
                                </div>

                                <div class="form-outline form-white mb-4">
                                    <input type="password" name="password" class="form-control form-control-lg" />
                                    <label class="form-label" for="typePasswordX">Password</label>
                                </div>

                                <p class="small mb-5 pb-lg-2"><a class="text-white-50" href="#">Forgot
password?</a></p>

                                <button class="btn btn-outline-light btn-lg px-5" type="submit">Login</button>
                            </form>

                            <div class="d-flex justify-content-center text-center mt-4 pt-1">
                                <a href="#" class="text-white"><i class="fab fa-facebook-f fa-lg"></i></a>

```



```

        <link
href="https://fonts.googleapis.com/css2?family=PT+Mono&family=Roboto:wght@300&display
=swap" rel="stylesheet">
        <!-- CSS only -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
0evHe/X+R7YkIZDRvuzKMRqM+OrBnVFBL6DOitfPri4tjfHxaWutUpFmBp4vmVor"
crossorigin="anonymous">
        <meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0, user-scalable=0">
        </head>
        <body>
<section class="vh-100 gradient-custom">
<div class="container py-5 h-100">
<div class="row d-flex justify-content-center align-items-center h-100">
<div class="col-12 col-md-8 col-lg-6 col-xl-5">
<div class="card bg-dark text-white" style="border-radius: 1rem;">
<div class="card-body p-5 text-center">

<div class="mb-md-5 mt-md-4 pb-5">

<h2 class="fw-bold mb-2 text-uppercase">Registration</h2>
<form action="/register" method="POST">
<p class="text-white-50 mb-5">Please enter your login and password!</p>

<div class="form-outline form-white mb-4">
<input type="email" name="username" class="form-control form-control-lg" />
<label class="form-label" for="typeEmailX">Email</label>
</div>

<div class="form-outline form-white mb-4">
<input type="password" name="password" class="form-control form-control-lg" />
<label class="form-label" for="typePasswordX">Password</label>
</div>

```



```

SMTP_USERNAME = "andreysydoruk@yahoo.com"
SMTP_PASSWORD = "xoisugficsjmhsju"
EMAIL_FROM = "andreysydoruk@yahoo.com"
EMAIL_TO = "ndk0011@gmail.com"
EMAIL_SUBJECT = "REMINDER:"

def send_email(co_msg):
    msg = MIMEText(co_msg)
    msg['Subject'] = EMAIL_SUBJECT + "Callback"
    msg['From'] = EMAIL_FROM
    msg['To'] = EMAIL_TO
    debuglevel = True
    mail = smtplib.SMTP(SMTP_SERVER, SMTP_PORT)
    mail.set_debuglevel(debuglevel)
    mail.starttls()
    mail.login(SMTP_USERNAME, SMTP_PASSWORD)
    mail.sendmail(EMAIL_FROM, EMAIL_TO, msg.as_string())
    mail.quit()

app = Flask(__name__)
app.secret_key = 'iknowyoucanseethis'
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///test.db'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
db = SQLAlchemy(app)

class User(db.Model):
    """ Create user table """
    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(80), unique=True)
    password = db.Column(db.String(80))
    courses = db.Column(db.String(80))
    fio = db.Column(db.String(240))

    def __init__(self, username, password, fio):

```

```

self.username = username
self.password = password
self.courses = "None"
self.fio = fio

@app.route('/')
@app.route('/home')
@app.route('/index',methods=['GET', 'POST'])
def home(methods=['GET', 'POST']):
    if request.method == 'GET':
        return render_template("index.html")
    else:
        co_msg = f"""
Hello, {request.form['name']}! Just wanted to callback
Email: {request.form['email']}
Phone: {request.form['phone']}
"""
        send_email(co_msg)
        return redirect(url_for('home'))

@app.route('/login', methods=['GET', 'POST'])
def login():
    if not session.get('logged_in'):
        """Login Form"""
    if request.method == 'GET':
        return render_template('signin.html')
    else:
        name = request.form['username']
        passw = request.form['password']
        try:
            data = User.query.filter_by(username=name, password=passw).first()
            if data is not None:

```



```

        session['logged_in'] = True
        return redirect(url_for('home'))
    else:
        return 'Dont Login'
except:
    return "Dont Login"
else:
    data = User.query.filter_by(username=session['user']).first()
    return render_template('profile.html')

@app.route('/register', methods=['GET', 'POST'])
def register():
    if request.method == 'POST':
        new_user = User(
            username=request.form['username'],
            password=request.form['password'],
            fio = request.form['fullname'])
        db.session.add(new_user)
        db.session.commit()
        session['user'] = request.form['username']
        session['fio'] = request.form['fullname']
        session['courses'] = "None"
        return render_template('signin.html')
    return render_template('signup.html')

@app.route("/logout")
def logout():
    """Logout Form"""
    session['logged_in'] = False
    return redirect(url_for('home'))

@app.route("/courses")
def courses():

```

```
return render_template("courses.html")
```

```
@app.route("/python")
```

```
def python():
```

```
    if not session.get('logged_in'):
```

```
        flash('Error please logged in')
```

```
    else:
```

```
        session['courses'] = "Python"
```

```
        co_msg = f"{session['user']} choose Python course"
```

```
        send_email(co_msg)
```

```
        return redirect(url_for('home'))
```

```
@app.route("/js")
```

```
def js():
```

```
    if not session.get('logged_in'):
```

```
        flash('Error please logged in')
```

```
    else:
```

```
        session['courses'] = "js"
```

```
        co_msg = f"{session['user']} choose js course"
```

```
        send_email(co_msg)
```

```
        return redirect(url_for('home'))
```

```
@app.route("/cpp")
```

```
def cpp():
```

```
    if not session.get('logged_in'):
```

```
        flash('Error please logged in')
```

```
    else:
```

```
        session['courses'] = "cpp"
```

```
        co_msg = f"{session['user']} choose cpp course"
```

```
        send_email(co_msg)
```

```
        return redirect(url_for('home'))
```

```
if __name__ == '__main__':  
    app.debug = True  
    db.create_all()  
    app.run(host='0.0.0.0',port=5000,use_reloader=True)
```


ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ

Ім'я файлу	Опис
Пояснювальні документи	
Кваліфікаційна робота_Філіппюк.doc	Пояснювальна записка до кваліфікаційної роботи. Документ Word.
Кваліфікаційна робота_Філіппюк.pdf	Пояснювальна записка до кваліфікаційної роботи в форматі PDF.
Програма	
diplom.zip	Архів. Містить коди програми.
Презентація	
Презентація_Філіппюк.ppt	Презентація кваліфікаційної роботи.