Міністерство освіти і науки України

Національний технічний університет

«Дніпровська політехніка»

_____Інститут електроенергетики_____

(інститут)

_____факультет інформаційних технологій_____

(факультет)

Кафедра__ інформаційних технологій та комп'ютерної інженерії ___

(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА

кваліфікаційної роботи ступеня ____бакалавра___

(бакалавра, спеціаліста, магістра)

Студента _____Кривенко Микита Андрійович_____

(ПІБ)

академічної групи _____126-19-1_____

(шифр)

спеціальності _____126 « Інформаційні системи та технології»_____

(код і назва спеціальності)

за освітньо-професійною програмою _____

_____ «Інформаційні системи та технології»_____

(офіційна назва)

на тему Використання фреймворку Node.js для створення веб чату з забезпеченням розширених можливостей користувачів

(назва за наказом ректора)

Керівники	**Прізвище, ініціали**	**Оцінка за шкалою**		**Підпис**
		рейтинговою	**інституційною**	
кваліфікаційної роботи	проф. Нікулін С.Л.			
розділів:				

Рецензент	Доцент Ширін А.Л			

Нормоконтролер	проф. Коротенко. Г.М.			

Дніпро

2023

ЗАТВЕРДЖЕНО:
завідувач кафедри
____інформаційних технологій____
___та комп'ютерної інженерії_____
(повна назва)
_____Гнатушенко В.В.__
(підпис) (прізвище, ініціали)

«_____»_____ 2023 року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня _____бакалавра_____
(бакалавра, спеціаліста, магістра)

студенту _____Кривенко М.А.___ **академічної групи** _____126-19-1____
(прізвище та ініціали) (шифр)

спеціальності _____126«Інформаційні системи та технології »_____

за освітньою-професійною програмою_____

_____«Інформаційні системи та технології»_____

на тему__Використання фреймворку Node.js для створення веб чату з забезпеченням розширених можливостей користувачів_____

затверджену наказом ректора НТУ «Дніпровська політехніка» від 16.05.2023 № 350-с

Розділ	**Зміст**	**Термін виконання**
Розділ 1	АНАЛІЗ СТАНУ ОБЛАСТІ РІШЕННЯ ЗАДАЧІ	10.02.2023 – 29.02.2023
Розділ 2	ПРОЕКТНІ РІШЕННЯ	04.03.2023 – 26.04.2023
Розділ 3	РЕАЛІЗАЦІЯ ПРОЕКТНИХ РІШЕНЬ	16.04.2023 – 14.05.2023

Завдання видано _____ ____Нікулін С.Л.___
(підпис керівника) (прізвище, ініціали)

Дата видачі _____01.02.2023 р._____

Дата подання до екзаменаційної комісії ____20.06.2023 р.____

Прийнято до виконання _____ ___Кривенко М.А.___
(підпис студента) (прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 92 стор.,18 мал. 3 додатка, 15 джерел, 3 - таб.

Об’єкт розроблення: Використання фреймворку Node.js для створення веб чату з забезпеченням розширених можливостей користувачів.

Мета кваліфікаційної роботи: Створення веб-чату для забезпечення зручного та ефективного спілкування між користувачами в режимі реального часу з допомогою Node.js, PHP, MySQL та jQuery

У вступі подані причини створення веб-чату для суспільства, короткий опис поняття “веб-чат”, значення FrontEnd`у та BackEnd`у та їх інструментів для розробки.

У першому розділі наведені технологій, які будуть використані для виконання цілей кваліфікаційної роботи. У роботі використовуються сучасні методи та технології для роботи над веб-сервісом.

У другому розділі наведені технічні завдання для роботи. Вимоги до функціональних характеристик, підстави для розробки.

У третьому розділі представлена реалізація проектних рішень, а саме:. Створення таблиць в БД, приклади таблиць та схеми. Архітектурна схема сайту та більш детальний опис виконаної кваліфікаційної роботи.

Практичне значення кваліфікаційної роботи полягає у тому що, створений веб-чат допомагає людям у! наданні інформаційних послуг!, детального опису послуг, немає необхідності користуватись друкарнями, доступ до сайту можливо отримати 24/7.

Ключові слова: ВЕБ-ЧАТ, JQUERY, СУБД, GIT, FRONTEND, BACKEND,ФРЕЙМВОРК, БД, PHP, CSS

ABSTRACT

Explanatory note: 34 pages, 13 pictures. - appendices, - sources, - tab.

Development object: Using the Node.js framework to create a web chat with advanced user capabilities.

The purpose of the qualification work: Creation of a web chat to provide convenient and effective communication between users in real time using Node.js, PHP, MySQL and jQuery

The introduction provides the reasons for creating a web chat for society, a brief description of the concept of "web chat", the meaning of FrontEnd and BackEnd, and their development tools.

The first section lists the technologies that will be used to fulfill the goals of the qualification work. The work uses modern methods and technologies for working on a web service.

In the second section, the technical tasks for the work are given. Requirements for functional characteristics, grounds for development.

The third section presents the implementation of project solutions, namely: Creating a repository in GitHub and assigning roles and tasks for development. Creating tables in the database, examples of tables and schemes. Architectural scheme of the site and a more detailed description of the completed qualification work.

The practical significance of the qualification work is that the created web chat helps people in! provision of information services!, detailed description of services, there is no need to use printers, access to the site can be obtained 24/7.

Keywords: WEBCHAT, JQWUERY, DBMS, GIT, FRONTEND, BACKEND, FRAMEWORK, DB, PHP, CSS

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

HTML - extensible hypertext markup language (розширювана мова гіпертекстової розмітки)

БД - баз даних

СУБД - системи управління базами даних

GIT - Global Information Tracker (система управління версіями)

ASGI - Asynchronous Server Gateway Interface (клиент-серверный протокол взаимодействия веб-сервера и приложения)

Веб-сервер - сервер, що зв'язують вхідні http запити з програмною частиною сайту.

Фронтенд - публічна частина веб сайту.

Бекенд - програмно-апаратна частина проекту.

ВСТУП

Студент, який обирає напрям підготовки "126 - Інформаційні системи та технології", згідно освітньо-професійної програми (ОПП) та освітньо-кваліфікаційної характеристики (ОКХ), має набути навичок у проектуванні, організації та реалізації функцій для доступу до баз даних (БД). В рамках цього курсу студент отримає практичні навички у розробці веб-проекту з використанням системи управління базами даних (СУБД).

Навіщо суспільству веб-чат?

1. Забезпечення ефективної комунікації: Веб-чат дозволяє користувачам спілкуватись один з одним в режимі реального часу з будь-якого місця з доступом до Інтернету. Це може бути особливо корисним у випадку важливих та невідкладних питань, коли потрібно швидко взяти рішення.

2. Зменшення витрат на спілкування: Веб-чат може знизити витрати на спілкування, оскільки користувачі можуть обмінюватись повідомленнями без необхідності витрачати кошти на дзвінки або поїздки.

3. Забезпечення доступності: Веб-чат може бути корисним для людей з різними видами обмежень, таких як люди з інвалідністю або ті, хто не може виходити з дому через обставини. Веб-чат дозволяє їм спілкуватись з іншими людьми та брати участь у суспільному житті.

4. Забезпечення швидкості та зручності: Веб-чат може бути дуже зручним та швидким засобом спілкування, оскільки повідомлення можуть бути відправлені та отримані миттєво з будь-якого місця з доступом до Інтернету.

Таким чином, метою виконання даного дипломного проекту є розробка веб-чату.

Backend сайту розроблений на PHP. За допомогою веб-чату можливо шукати, вибирати користувачів, передивлятись які користувачі зараз в онлайні, а також зберігати повідомленя.

Сайт має Frontend, розроблений з використанням HTML, CSS, jQuery, який допомагає спростити та поліпшити роботу з JavaScript на стороні клієнта. jQuery

надає розробникам можливість легко та ефективно взаємодіяти з HTML-документами, керувати DOM-елементами, виконувати анімацію та обробляти події користувача.

РОЗДІЛ 1
АНАЛІЗ СИТУАЦІЇ ТА СТАНУ ГАЛУЗІ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕННОЇ ЗАДАЧІ.

1.1 АКТУАЛЬНІСТЬ ТЕМИ

Нині цифрова комунікація стала невід'ємною частиною нашого повсякденного життя. Розвиток інтернет-технологій призвело до виникнення та розповсюдження різноманітних засобів комунікації, включаючи електронну пошту, соціальні мережі, месенджери та веб-чати. Веб-чати, зокрема, є засобами онлайн-спілкування, що дозволяють користувачам обмінюватися текстовими повідомленнями в режимі реального часу.

Сьогоднішній світ став більш пов'язаним та мобільним, і веб-чати стали невід'ємною складовою нашої цифрової комунікаційної палітри. Веб-чати відкривають перед нами нові горизонти у спілкуванні, дозволяючи встановлювати контакти, ділитися інформацією та підтримувати взаємодію з іншими людьми по всьому світу. Вони надають можливість вільного обміну ідеями, думками, знаннями та емоціями, незалежно від фізичної відстані.

Веб-чати стали невід'ємною частиною нашої культури та соціального життя. Вони проникли у всі сфери нашого суспільства, включаючи бізнес, освіту, охорону здоров'я та розваги. Завдяки своїй простоті використання та доступності, веб-чати приваблюють все більше людей, включаючи різні вікові групи та соціальні верстви. Вони стали невід'ємним інструментом у створенні робочого процесу, навчання, торгівлі та особистого спілкування.

Таким чином, веб-чати є актуальним об'єктом дослідження у сучасній епосі цифрової комунікації.

1.2 ВАРІАНТИ РОЗРОБКИ ДЛЯ FRONTEND`У

1.2.1 ПОНЯТТЯ FRONT END

Фронтенд представляє собою публічну складову веб-додаток, яка дозволяє швидко взаємодіяти користувачів з ними. Вона включає відображення користувацьких функцій та інтерфейсу, які працюють на боці клієнта, а також обробку запитів користувачів. Фронтенд - це те, що спостерігає користувача при відкритих веб-сторінках.

Веб-додаток є клієнт-серверним застосунком, де браузер діє як клієнт, а веб-сервер - як сервер. Логіка додатка розподіляється між сервером і клієнтом, дані зберігаються переважно на сервері, а обмін відомостями здійснюється через мережу. Іншими словами, це те, що користувач бачить і робить кожного разу, коли він підключається до Інтернету та відкриває браузер.

Frontend-розробка охоплює створення публічної частини веб-додатка, з якою користувач взаємодіє, а також функціоналу, який виконується на стороні клієнта. Розробник фронтенду працює над тим, щоб кожна кнопка, іконка, текст і вікно на веб-сайті були розміщені в належному порядку, не перекривали один і малий згодний вигляд. Крім того, вони повинні отримати ваше призначення, наприклад, кнопка "Відправити" має надіслати повідомлення.[1]

1.2.2 ІНСТРУМЕНТИ ДЛЯ FRONT END’y

На сьогоднішній день існує широкий спектр інструментів та фреймворків для розробки фронтенду. Проте, є три основних компоненти, які є невід'ємною частиною будь-якого веб-сайту.

А. HTML (HyperText Markup Language) - це мова, яка використовується для створення структури веб-сторінок. За допомогою HTML можна організовувати текст у вигляді абзаців та заголовків, створювати списки, таблиці, а також вставляти графічні зображення, аудіо та відео на сторінку.[2]

Б. CSS (Cascading Style Sheets) - це мова, яка відповідає за зовнішній вигляд веб-сторінок. Вона дозволяє встановлювати стиль шрифтів для тексту абзаців та заголовків, задавати фоновий колір навігаційної панелі, визначати рамки навколо елементів, а також змінювати колір гіперпосилань при наведенні курсору миші, створюючи тим самим різноманітні анімаційні ефекти.[3]

В. JavaScript - це мова програмування, яка використовується для надання веб-сторінкам інтерактивності. Вона підтримує різні парадигми програмування, зокрема об'єктно-орієнтований, імперативний та функціональний підходи. JavaScript широко використовується у веб-браузерах для створення динамічних та інтерактивних елементів на веб-сторінках.[4]

1.2.3 ФРЕЙМВОРК:ЙОГО ПОНЯТТЯ ТА ПЕРЕВАГИ

Фреймворк - це платформа, яка надає розробникам базовий набір інструментів для створення програмного забезпечення. Він включає попередньо визначені класи або функції, які можна використовувати. Крім того, для вирішення конкретних завдань можна додавати власний код до існуючих можливостей фреймворку. [5]

Основні переваги використання фреймворків:

- Підвищена продуктивність: фреймворки суттєво полегшують і прискорюють розробку шляхом використання оптимізованого коду та добре організованих шаблонів, замість написання сотень рядків коду.

- Економічність: фреймворки є безкоштовними та мають відкритий вихідний код, що дозволяє зменшити витрати на розробку веб-додатків. Вони також сприяють швидкому розвитку додатків, що знижує загальну вартість проекту.

- Безпека: сучасні JavaScript-фреймворки мають значну підтримку великою спільнотою на GitHub та надійну систему безпеки.

1.2.4 МОЖЛИВОСТІ CSS ТА ЙОГО ФРЕЙМВОРКІВ BOOTSTRAP. FOUNDATION. TAILWIND CSS.

Для проекту я використовував звичайний CSS і зараз розповім чому саме. Основною перевагою використання звичайного CSS над фреймворками є те, що він дає можливість більш точно контролювати вигляд та поведінку елементів на сторінці. На відміну від фреймворків, які надають заздалегідь визначені класи та стилі для елементів, CSS дає розробнику повну свободу визначати стилі для кожного елемента окремо. Це означає, що розробник може дуже точно контролювати вигляд та поведінку кожного елемента на сторінці, що може бути особливо важливим для деяких проектів, де потрібно забезпечити високу точність дизайну.

Крім того, звичайний CSS використовується у більшості веб-проектів, тому для більшості розробників це є зрозумілим та звичним інструментом роботи, що дає можливість швидко та ефективно розробляти веб-сторінки без необхідності вивчення нових інструментів та технологій.

Хоча фреймворки CSS, такі як Bootstrap, Materialize, Foundation та інші, можуть допомогти розробникам значно скоротити час розробки та забезпечити швидкий та

ефективний розгортання веб-сторінок з високоякісним дизайном, використання звичайного CSS може бути більш підходящим варіантом для проектів, які вимагають високої точності та контролю вигляду та поведінки кожного елемента на сторінці.

Давайте проведемо огляд відомих CSS-фреймворків, які використовуються для розробки фронтенду FrontEnd`у: [6]

1) Bootstrap є безкоштовним набором інструментів для розробки веб-сайтів і додатків. Він включає готові HTML- і CSS-шаблони для стилізації типографіки, веб-форм, кнопок, міток, навігаційних блоків та інших компонентів веб-інтерфейсу. Крім того, Bootstrap має JavaScript-розширення. [7] На рисунку 1 показано приклад логотипу.

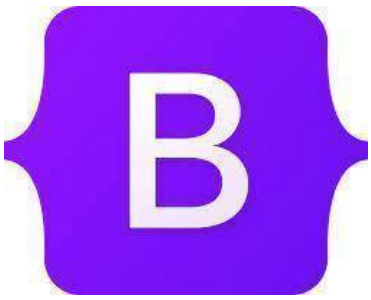


Рис. 1. Логотип Bootstrap

Bootstrap 5 - це визнана бібліотека компонентів, що набула популярності. Вона була випущена у 2021 році та пропонує ще більше зручних утиліт і модифікаторів. Bootstrap 5 був написаний на SASS, що дозволяє використовувати всі переваги препроцесорів у процесі розробки.

2) Foundation - це інтуїтивний інтерфейс, який пропонує адаптивну сітку та компоненти HTML і CSS для користувацького інтерфейсу. Він включає шаблони, фрагменти коду, типографіку, форми, кнопки, навігацію та інші елементи інтерфейсу. Крім того, Foundation надає додаткові можливості, такі як JavaScript-розширення. [8] На рисунку 2 показано приклад логотипу.

Рис. 2. Логотип Foundation

Це рішення є вишуканим і гнучким, ідеальним для великих проектів. Він використовується такими компаніями, як Facebook, eBay, Mozilla, Adobe, HP, Cisco та Disney.

3)Tailwind CSS - це інноваційний фреймворк CSS, який пропонує унікальний підхід до стилізації елементів. Замість фокусу на функціональності окремих елементів, Tailwind ставить акцент на те, як їх відображати. Це дозволяє розробникам легко тестувати нові стилі і змінювати макет. [9] На рисунку 3 показаний приклад логотипу. Tailwind CSS використовується як утиліта першого фреймворка CSS.

Рис. 3 Логотип Tailwind CSS

Цей фреймворк низького рівня надає можливість повної кастомізації. Tailwind CSS є ідеальним вибором для нестандартних дизайнерських рішень.

1.2.5 ЩО ТАКЕ CSS

CSS (Cascading Style Sheets) - це мова, що використовується для опису вигляду та форматування веб-сторінок. CSS визначає, які кольори, шрифти, розміри та

розташування елементів на сторінці. CSS використовується разом з HTML для створення привабливих та добре оформлених веб-сторінок.

Основні інструменти Bootstrap:

• Селектори: Це патерни, що вказують, які HTML елементи повинні бути оформлені згідно з CSS стилями.

• Властивості: Це параметри, що визначають вигляд та поведінку HTML елементів, такі як колір, розмір, положення та інші.

• Значення: Це значення, які призначені для властивостей, такі як кольори, розміри та інші параметри.

• Коментарі: Це спеціальні рядки тексту, які не будуть відображені у вихідному коді сторінки, але можуть бути корисні для розробників, які переглядають код.

• Медіа-запити: Це спосіб, який дозволяє налаштувати стилі, щоб вони працювали оптимально на різних пристроях та екранах.

• Розташування: Це вказівки для розташування елементів на сторінці, такі як блочна модель, Flexbox та CSS Grid.

1.3 ВАРІАНТИ ІНСТРУМЕНТІВ ДЛЯ РОЗРОБКИ BACKEND-СИСТЕМИ.

Нині доступні різноманітні інструменти для створення веб-сайтів. Ви можете розпочати розробку з нуля, скористатися CMS-системами, такими як WordPress, Joomla, Drupal, vBulletin, або використовувати бекенд-фреймворки, такі як NodeJS, Django, Laravel та інші.

Завданням даної дипломної роботи є розробка швидкого веб-чату в реальному часі з обмеженим функціоналом, що забезпечує безпеку розробки і легкість масштабування для майбутніх змін і розширень. У результаті ретельного аналізу, я вирішив використати гнучку мову програмування PHP для цього проекту.

1.3.1 ПОНЯТТЯ BACKEND

Backend - це складова програмна система, розташована на серверній стороні комплексних веб-систем, що відрізняється від фронтенду. Він відповідає за управління даними проекту, забезпечує доступ до них, здійснює модифікації та передає відповідні дані до клієнтської сторони. Backend також часто називають "двигуном" веб-сайту, оскільки він забезпечує роботу всіх необхідних процесів, які забезпечують функціональність веб-додатка. [13]

1.4 ОБРАННЯ СУБД

1.4.1 ПОНЯТТЯ СУБД

Система управління базами даних (СУБД) є комплексним рішенням, що включає в себе набір взаємопов'язаних даних, включаючи бази даних, а також програми, які забезпечують доступ до цих даних. [10] СУБД забезпечує можливості створення, збереження, оновлення та пошуку інформації в базах даних, здійснюючи при цьому контроль доступу до даних.

1.4.2 УПРАВЛІННЯ БД (СУБД). MYSQL. MongoDB. PostgreSQL

При розробці архітектури додатку, вибір системи управління баз даних є критичним етапом. Вона повинна задовольняти різноманітні вимоги додатку, такі як швидкість операцій, безпека зберігання, доступність через мережу Інтернет, наявність необхідного набору функцій, фінансові витрати на розробку та придбання обладнання та програмного забезпечення, а також необхідність навчання персоналу.

Відомі СУБД:

1) MySQL є безкоштовною системою керування реляційними базами даних, розробленою компанією "TcX" з метою покращення продуктивності обробки великих обсягів даних. [11] Ця відкрита СКБД створена як альтернатива комерційним системам. Початково MySQL була схожою на mSQL, але з часом вона значно розширилася і стала однією з найпопулярніших СКБД. Вона широко використовується для розробки динамічних веб-сторінок, оскільки має відмінну підтримку для різноманітних мов програмування.

2) **MongoDB -** це відкрита система керування базами даних, що орієнтується на роботу з документами, і не вимагає опису схеми таблиць. Вона займає проміжну позицію між швидкими та масштабованими системами, що працюють з даними у форматі ключ-значення, і реляційними системами керування базами даних, надаючи потужні функціональні можливості та зручний спосіб формування запитів.[12]

3) **PostgreSQL**, також відома як «Пост-грес-К'ю-ель» або «постгрес», є об'єктно-реляційною системою керування базами даних (СКБД). Вона є альтернативою як комерційним СКБД, таким як Oracle Database, Microsoft SQL Server, IBM DB2 і інші, так і СКБД з відкритим кодом, наприклад MySQL, Firebird, SQLite.

У порівнянні з іншими проєктами з відкритим кодом, такими як Apache, FreeBSD або MySQL, PostgreSQL не контролюється жодною окремою компанією. Його розробка здійснюється завдяки співпраці багатьох людей та компаній, які хочуть використовувати цю СКБД і впроваджувати в неї останні досягнення.

Для тестування програми буде використана MariaDB, що є аналогом MySQL.

Для деплою на сервер буде використана СКБД phpMyAdmin, оскільки вона повністю задовольняє всі вимоги додатка.

1.5 ОБРАННЯ ХОСТИНГУ

Оскільки поточний дипломний проект не отримує фінансування, тому буде використаний дистрибутив для зборки локального веб-серверу - XAMPP

XAMPP – безкоштовний кросплатформовий дистрибутив для складання локального веб-сервера. Містить Apache, MariaDB, мову програмування Perl, інтерпретатор скриптів PHP та додаткові бібліотеки. Має відкритий вихідний код, простий у встановленні та використанні.[14]

Рис. 4 Логотип XAMPP

Назва є акронімом:

X - кросплатформність, тобто збірка підходить для всіх ОС;

A - веб-сервер Apache2;

M - база даних MariaDB;

P – інтерпретатор мови PHP;

P - Strawberry Perl, одна з реалізацій мови Perl.

XAMPP дозволяє навіть новачкові швидко розгорнути веб-сервер на будь-якій операційній системі без фінансових витрат. Також розробники можуть використовувати складання для тестування роботи веб-сторінок та сайтів до внесення змін до основного проекту.

Переваги XAMPP:

• Легкість встановлення та налаштування: XAMPP дозволяє швидко та легко встановити та налаштувати Apache, MySQL, PHP та Perl на локальному комп'ютері без необхідності окремого встановлення кожного компонента.

• Переносність: XAMPP може бути легко перенесений з одного комп'ютера на інший, що дозволяє розробникам працювати зі своїми проектами на різних машинах.

• Легкий доступ до файлів та баз даних: XAMPP забезпечує легкий доступ до файлів та баз даних, що дозволяє розробникам швидко та легко редагувати та тестувати свої проекти.

• Підтримка багатьох операційних систем: XAMPP забезпечує підтримку різних операційних систем, включаючи Windows, Linux та Mac OS X, що дозволяє користувачам використовувати його функціональність на різних платформах.

• Безкоштовний та відкритий код: XAMPP є безкоштовним та відкритим дистрибутивом, що дозволяє розробникам економити час та гроші на розгортанні своїх проектів.

Саме посилання на цей сайт - https://www.apachefriends.org/ru/index.html

1.6 ОБИРАННЯ СИСТЕМИ УПРАВЛІННЯ ВЕРСІЯМИ

1.6.1 АНАЛІЗ КОНЦЕПЦІЇ СИСТЕМИ КОНТРОЛЮ ВЕРСІЙ GIT

Git є розподіленою системою керування версіями файлів і спільної роботи, розробленою Лінусом Торвальдсом для управління розробкою ядра Linux. На сьогоднішній день ця система підтримується Джуніо Хамано. Git вважається однією з найефективніших, надійних і продуктивних систем керування версіями, що надає гнучкі засоби для нелінійної розробки на основі гілок, здатність до злиття різних гілок. Щоб забезпечити цілісність історії та стійкість до змін, використовуються криптографічні методи, а також можлива прив'язка цифрових підписів розробників до тегів і комітів.

1.6.2 АНАЛІЗ СИСТЕМИ УПРАВЛІННЯ ВЕРСІЯМИ - GITHUB.

GitHub - це один з найбільших веб-сервісів для спільної розробки програмного забезпечення. Платформа пропонує як безкоштовні, так і платні тарифні плани для

користувачів. [15] Вона ґрунтується на системі керування версіями Git і розроблена компанією GitHub, Inc. (раніше відомою як Logical Awesome) з використанням Ruby on Rails і Erlang.

Безкоштовний доступ надається для проєктів з відкритим вихідним кодом, що включає всі можливості, включаючи SSL-шифрування, а для окремих індивідуальних проєктів доступні різні платні тарифні плани.

Рис. 5 Логотип GitHub

1.7 МЕТА ТА ЗАДАЧА ВЕБ-ЧАТУ “SYTLE”

Метою мого веб-чату є створення ефективного засобу комунікації, який сприятиме покращенню взаємодії між людьми у цифровій епосі. Ми прагнемо надати користувачам можливість швидкого, зручного та безпечного обміну інформацією в режимі реального часу, долаючи географічні обмеження.

Головною задачою вважається розробка інтуїтивно зрозумілого інтерфейсу користувача, який буде легко зрозумілий і зручний у використанні навіть для непрофесійних користувачів. Ми прагнемо простоти та інтуїтивної навігації, щоб люди могли швидко освоїти функціональність веб-чату і почати використовувати його без труднощів.

1.8 ВИСНОВКИ ДО ПЕРШОГО РОЗДІЛУ

У цьому розділі розглянуто різні аспекти, пов'язані з розробкою веб-чатів. Починаючи з актуальності теми, було зазначено, що веб-чати стали невід'ємною

частиною сучасного інтернету та активно використовуються для спілкування та взаємодії користувачів. Були розглянуті варіанти розробки для front-end, представлені можливості CSS та його фреймворків, таких як Bootstrap, Foundation та Tailwind CSS. CSS є мовою стилів, яка використовується для оформлення та візуалізації веб-сторінок.

Далі було розглянуто варіанти інструментів розробки backend-системы. Було наголошено на важливому виборі СУБД (системи управління базами даних), який залежить від вимог проекту та особливостей веб-чату. Також було розглянуто вибір хостингу, важливого аспекту при розгортанні веб-чату, який має забезпечувати високу доступність та продуктивність системи.

Нарешті, було визначено мету та завдання веб-чату. Метою може бути забезпечення ефективної взаємодії користувачів, надання зручного інтерфейсу для обміну інформацією або вирішення конкретних завдань. Завдання веб-чату визначає його функціональність, включаючи можливість надсилання повідомлень, створення кімнат для спілкування, автентифікацію користувачів та інші особливості, залежно від поставлених завдань.

Загалом розділ присвячений веб-чатам представляє огляд актуальності даної теми, розгляд варіантів розробки для front-end і back-end, вибору необхідних інструментів, а також визначення мети та завдання веб-чату. Ця інформація є важливою для подальшої розробки та реалізації веб-чату, забезпечуючи розуміння основних аспектів та напрямків роботи.

РОЗДІЛ 2.
РІШЕННЯ ПРОЕКТУ

2.1 ЗАВДАННЯ НА РОЗРОБКУ ПРОГРАМНОГО ПРОДУКТУ

2.1.1. АНОТАЦІЯ ТА СФЕРА ВИКОРИСТАННЯ

Веб-чат “Sytle” з вирокастанням розширених можливостей користувачів.

2.1.2 ОСНОВИ РОЗРОБКИ

Дипломний проект виконується відповідно до наказу ректора НТУ «Дніпровська політехніка» від 16.05.2023 № НАКАЗ 350-с

2.1.3 ЦІЛЬ РОЗРОБКИ

Веб-чат служить засобом комунікації, підтримки та соціального спілкування.

2.1.4 ФУНКЦІОНАЛЬНІ ВИМОГИ ДО ХАРАКТЕРИСТИК

Функціональні властивості веб-сайту повинні гарантувати наступні аспекти:

• Обмін повідомленнями у реальному часі.

• Функціональність, що дозволяє додавати, змінювати та видаляти користувачів з використанням бази даних.

• Відображеня статуса користувача (онлайн/оффлайн)

• Можливість пошуку користувачів по нікнейму або email

• Відображеня часу повідомлення та дати листування

2.1.5. ВИМОГИ ДО НАДІЙНОСТІ СИСТЕМИ

Безпека і надійність веб-сайту залежать від таких факторів:

• Захист від несанкціонованого доступу

• Захист від вразливостей

• Резервне копіювання даних

• Висока доступність

• Захист від шкідливих атак

2.1.6. ТЕХНІЧНІ ВИМОГИ ДО СКЛАДУ І ХАРАКТЕРИСТИК ТЕХНІЧНИХ ЗАСОБІВ

Для ефективної роботи сервера необхідно забезпечити наступні компоненти та умови:

• Процесор: Мінімально необхідний процесор з достатньою швидкістю для обробки запитів і завдань сервера. Це може бути процесор з одним або більше ядрами, залежно від потреб вашого серверного програмного забезпечення.

• Оперативна пам'ять (RAM): Достатня кількість оперативної пам'яті для виконання операційної системи сервера та запущених на ньому додатків. Мінімальний обсяг RAM може варіюватися від кількох гігабайт до десятків гігабайт, в залежності від потреб вашого серверного програмного забезпечення і обсягу даних, з якими воно працює.

• Простір на жорсткому диску: Достатній обсяг простору на жорсткому диску для збереження операційної системи, додатків, конфігураційних файлів і даних, які обробляє сервер. Розмір залежить від ваших потреб, але часто рекомендується мати запасний простір для майбутнього зростання.

• Мережеві інтерфейси: Сервер повинен мати мережеві інтерфейси для забезпечення з'єднання з мережею і доступу до сервера через мережу. Це може бути Ethernet-порти або бездротові інтерфейси з підтримкою відповідних протоколів.

• Безперебійне живлення (UPS): Для забезпечення безперебійного функціонування сервера рекомендується використовувати UPS, що забезпечує резервне живлення у разі відключення основного джерела електроживлення.

2.1.7. ВИМОГИ ДО СУМІСНОСТІ ІНФОРМАЦІЙНИХ ТА ПРОГРАМНИХ КОМПОНЕНТІВ

Дана програма розроблена для використання на операційних системах Windows і сумісних з Unix / Linux.

2.1.8. ВИМОГИ ДО ПРОГРАМНОЇ ДОКУМЕНТАЦІЇ

Для веб-сайту, який був розроблений, необхідно надати наступну технічну документацію:

- Опис архітектури
- Функціональні вимоги
- API та інтеграції
- Бази даних
- Конфігурація та налаштування
- Інструкції розгортання
- Інструкції обслуговування
- Безпека

2.2 ВИСНОВКИ ДО ДРУГОГО РОЗДІЛУ

У даному розділі було розглянуто основні аспекти, пов'язані з проектними рішеннями. Завдання на розробку програмного продукту були визначені і описані, включаючи анотацію та сферу його використання. Було розглянуто основи розробки програмного продукту.

Ціль розробки була чітко визначена. Було визначено функціональні вимоги до характеристик програмного продукту, включаючи його основні функції та можливості. Також було визначено вимоги до надійності системи, включаючи вимоги до стійкості, доступності та безпеки програмного продукту.

У розділі були також розглянуті технічні вимоги до складу і характеристик технічних засобів, необхідних для роботи програмного продукту. Це включає апаратне забезпечення, мережеві засоби та інші компоненти, які необхідні для ефективної роботи системи.

Загальною метою розділу було забезпечити ясність і чіткість вимог і характеристик, пов'язаних з розробкою програмного продукту, щоб забезпечити його успішну реалізацію та ефективну роботу в майбутньому.

З урахуванням вищезазначеного, цей розділ відіграє важливу роль у плануванні та розробці програмного продукту, забезпечуючи основні вказівки та вимоги, які слід дотримуватись протягом усього процесу розробки та реалізації.

РОЗДІЛ 3.
РОЗРОБКА ПРОЕКТНИХ РІШЕНЬ

3.2 РОЗРОБКА ТА СТВОРЕННЯ ТАБЛИЦЬ

3.2.1 ПРИКЛАД ПРОЦЕСУ СТВОРЕННЯ ТАБЛИЦІ В БАЗІ ДАНИХ

PhpMyAdmin пропонує зручний веб-інтерфейс для ефективної роботи з базами даних MySQL, що дозволяє адміністраторам керувати базами даних через браузер.

Представлено наведений приклад таблиці користувачів на діаграмі під номером 6.

```
CREATE TABLE `users` (
  `user_id` int(11) NOT NULL,
  `unique_id` int(200) NOT NULL,
  `fname` varchar(255) NOT NULL,
  `lname` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `img` varchar(400) NOT NULL,
  `status` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Рис. 6

3.2.2 МОДЕЛЬ БАЗИ ДАНИХ

Система бази даних в проекті представлена на ілюстрації номер 7.

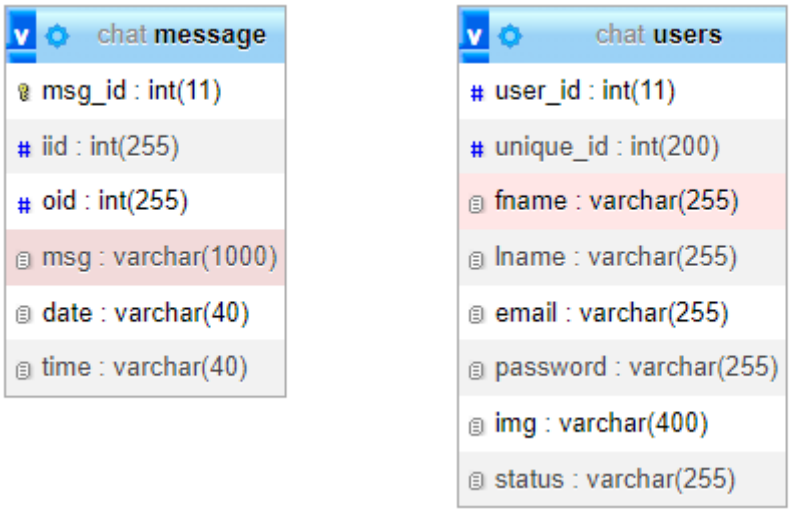


Рис. 7 Схема баз даних

3.2.3 ПЕРЕЛІК ТАБЛИЦЬ

Таблиця №1

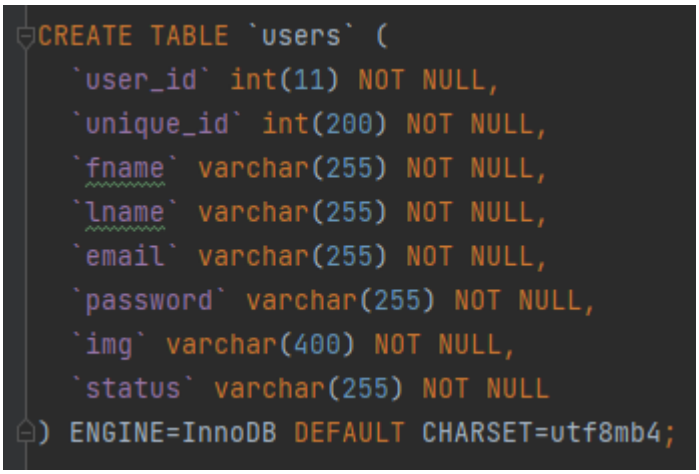
Перелік таблиць баз даних

№	Назва	Описання
1	users	Містить інформацію про користувачів, які стровані в системі
2	message	Зберігає в собі історію повідомлень

Таблиця №2

users

№	Назва поля	Вид	Обсяг	Ключ	Примітка
1	user_id	int	11	-	Загальний id користувача
2	unique_id	int	200	-	id сессії користувача
3	fname	varchar	255	-	Ім’я користувача
4	lname	varchar	255	-	Призвіще користувача
5	email	varchar	255	-	Email користувача
6	password	varchar	255	-	Пароль від аккаунта
7	img	varchar	400	-	Фотографія користувача
8	status	varchar	255	-	Статус (онлайн/оффлайн)



```
CREATE TABLE `users` (
  `user_id` int(11) NOT NULL,
  `unique_id` int(200) NOT NULL,
  `fname` varchar(255) NOT NULL,
  `lname` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `password` varchar(255) NOT NULL,
  `img` varchar(400) NOT NULL,
  `status` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

Рис. 8 Створення таблиці users

Таблиця - 3

message

№	Назва поля	Вид	Обсяг	Ключ	Примітка
1	msg_id	int	11	+	id повідомлення
2	iid	int	255	-	ідентифікатор жувача
3	oid	int	255	-	ідентифікатор равника
4	msg	varchar	1000	-	змінна для зберігання домлення
5	date	varchar	40	-	змінна для зберігання повідомлення
6	time	varchar	40	-	змінна для зберігання повідомлення

```sql
CREATE TABLE `message` (
  `msg_id` int(11) NOT NULL,
  `iid` int(255) NOT NULL,
  `oid` int(255) NOT NULL,
  `msg` varchar(1000) NOT NULL,
  `date` varchar(40) NOT NULL,
  `time` varchar(40) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

ALTER TABLE `message`
  ADD PRIMARY KEY (`msg_id`);

ALTER TABLE `message`
  MODIFY `msg_id` int(11) NOT NULL AUTO_INCREMENT;
COMMIT;
```

Рис. 9 процес створення таблиці для розсилки повідомлень.

3.3 АРХІТЕКТУРНИЙ ОГЛЯД ВЕБ-САЙТУ

3.3.1 ПРОЦЕС ПРОЕКТУВАННЯ БАЗИ ДАНИХ

phpMyAdmin має широкий спектр функцій для управління базами даних. Ви можете створювати, змінювати та видаляти бази даних, таблиці, поля, індекси та інші об'єкти бази даних. Також ви можете виконувати SQL-запити, імпортувати та експортувати дані, керувати користувачами та привілеями, а також багато іншого.

phpMyAdmin підтримує безліч мов, що дозволяє адаптувати інтерфейс до потреб різних користувачів.

```sql
create DATABASE IF NOT EXISTS chat CHARSET utf8mb4;
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";
use chat;
```

Рис. 10 Створення БД нашого веб-чату

Приклад методів:

• create() - Створює БД

• set - Оновлення значень стовпців у таблиці.

• use - Змінює поточну активну базу даних на вказану.

3.3.2 ПРЕДСТАВЛЕННЯ ВЕБ-СТОРІНКИ

На початку коду виконується функція session_start(), яка ініціалізує PHP сеанс. Потім відбувається увімкнення файлу php/config.php, який, містить налаштування бази даних та інші конфігурації.

Далі перевіряється, чи встановлена змінна сесії unique_id. Якщо її не встановлено, користувач перенаправляється на сторінку login.php за допомогою JavaScript-перенаправлення.

Потім слідує розмітка HTML-сторінки. Всередині <head> визначені мета-теги, посилання на файл CSS (style2.css) і заголовок сторінки.

У <body> визначено контейнер , у якому знаходиться розділ <section class="chat-area">, що містить чат.

У розділі <header> виконується PHP-код, який також включає файл php/config.php. Потім виходить значення параметра userid із рядка запиту та виконується SQL-запит до БД. Якщо SQL-запит повертає один або більше рядків, отримані дані використовуються для відображення інформації про користувача, такий як ім'я, фотографія та статус. Якщо SQL-запит не повертає жодного рядка, користувач перенаправляється на іншу сторінку за допомогою JavaScript-перенаправлення.

У розділі знаходиться блок з інформацією про користувача, включаючи його фотографію, ім'я та статус. Фотографія користувача виходить із бази даних.

Під розділом <header> розташований , який, призначений для відображення повідомлень чату.

Внизу сторінки знаходиться форма <form class="typing-area"> для введення нових повідомлень. Вона містить приховані поля outgoingchat і incomingchat, які зберігають ідентифікатори відправника та одержувача повідомлень відповідно. Користувач може ввести нове повідомлення в поле <input type="text" placeholder="Type a Message...." і відправити його, натиснувши кнопку <button class ="msgbtn">.

```php
<?php
include_once "php/config.php";
$userid = mysqli_real_escape_string($conn, $_GET['userid']);
$sql = mysqli_query($conn, query: "SELECT * FROM users WHERE unique_id = {$userid}");
if(mysqli_num_rows($sql) > 0){
    $row=mysqli_fetch_assoc($sql);
    $cssprop="";
    $imgsrc="";
    ($row['status']=="offline")?$cssprop="offline":$cssprop="";
    ($row['status']=="offline")?$imgsrc="pics/grey.png":$imgsrc="pics/greencircle.png";

}else{
    echo "<script> location.href='.php'; </script>";
}

?>
```

Рис. 11 Приклад роботи однієї з функції відображення веб-сторінки Main

3.3.3 ОПИС РОЗРОБЛЕНОГО САЙТУ І ЙОГО ІНТЕРФЕЙСУ

Головна сторінка веб-чату слугує центральним вузлом для спілкування користувачів у режимі реального часу. Вона має візуально привабливий і зручний

інтерфейс, що сприяє ефективній взаємодії. У верхній частині сторінки користувачі можуть знайти основну інформацію про свій профіль, таку як ім'я користувача, зображення профілю та онлайн статус. Крім того, зручно розташоване поле пошуку, що дозволяє користувачам швидко знаходити конкретних людей в чаті.

Безпосередньо під розділом інформації про користувача відображається повний список користувачів, що надає загальне уявлення про учасників чату. Кожен користувач представлений своїм іменем користувача разом із зазначенням його поточного статусу, незалежно від того, чи він онлайн, чи офлайн. Крім того, відображається останнє повідомлення, надіслане кожним користувачем, що дозволяє зазирнути в нещодавні розмови. Примітно, що коли користувач онлайн, його запис у списку виділяється яскравим фоном, що полегшує ідентифікацію активних учасників. І навпаки, рис. 12

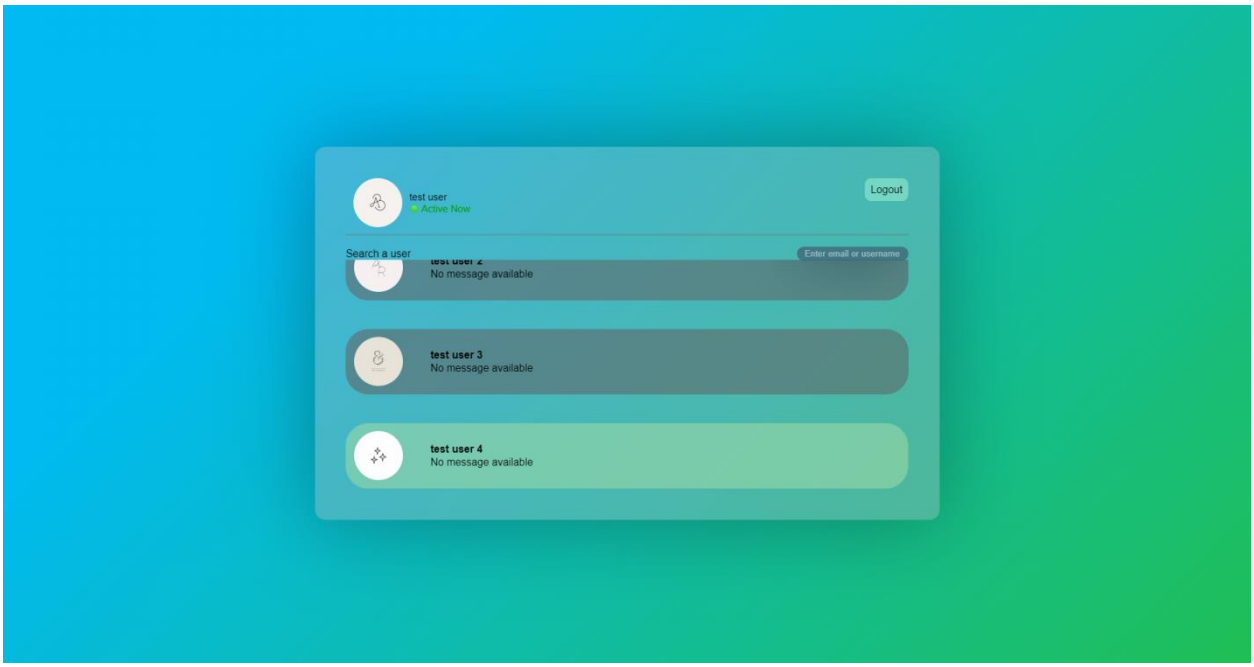


Рис. 12 Головна сторінка веб-чату

Щоб розпочати розмову з іншим користувачем, достатньо натиснути на його ім'я, щоб відкрити сторінку листування (Рис. 13).

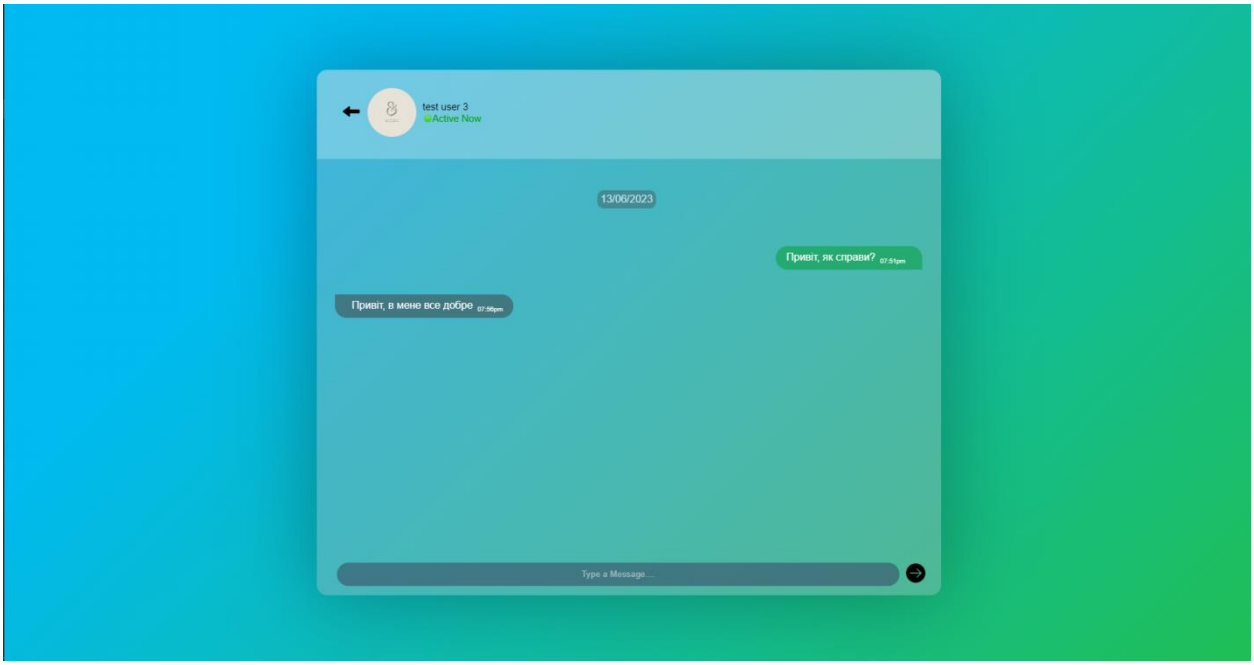


Рис. 13 Сторінка листування

Ця спеціальна сторінка демонструє безшовний та інтуїтивно зрозумілий дизайн, забезпечуючи цілеспрямоване середовище для обміну повідомленнями. На сторінці листування на видному місці відображається інформація про обраного користувача, включаючи його ім'я, фотографію профілю і позначку часу, що вказує на дату і час кожного повідомлення. Повідомлення, надіслані "нами", продумано виділяються яскравим фоном, що полегшує їх читання і забезпечує чіткість потоку розмови. На противагу цьому, повідомлення від інших користувачів представлені на темнішому фоні, що виокремлює їхній внесок.

Розроблений сайт та його інтерфейс мають пріоритетний підхід, орієнтований на користувача, підкреслюючи простоту, функціональність та привабливу візуальну естетику. Зручна навігація в поєднанні з чітким представленням інформації для користувачів та обміну повідомленнями сприяє ефективному спілкуванню в середовищі веб-чату. Завдяки інтуїтивно зрозумілому дизайну та увазі до деталей, сайт пропонує покращений користувацький досвід, полегшуючи змістовну взаємодію та сприяючи створенню активної онлайн-спільноти.

На сторінці реєстрації користувача (Рис. 14) представлені п'ять полей, що дозволяють ввести необхідну інформацію для створення облікового запису:

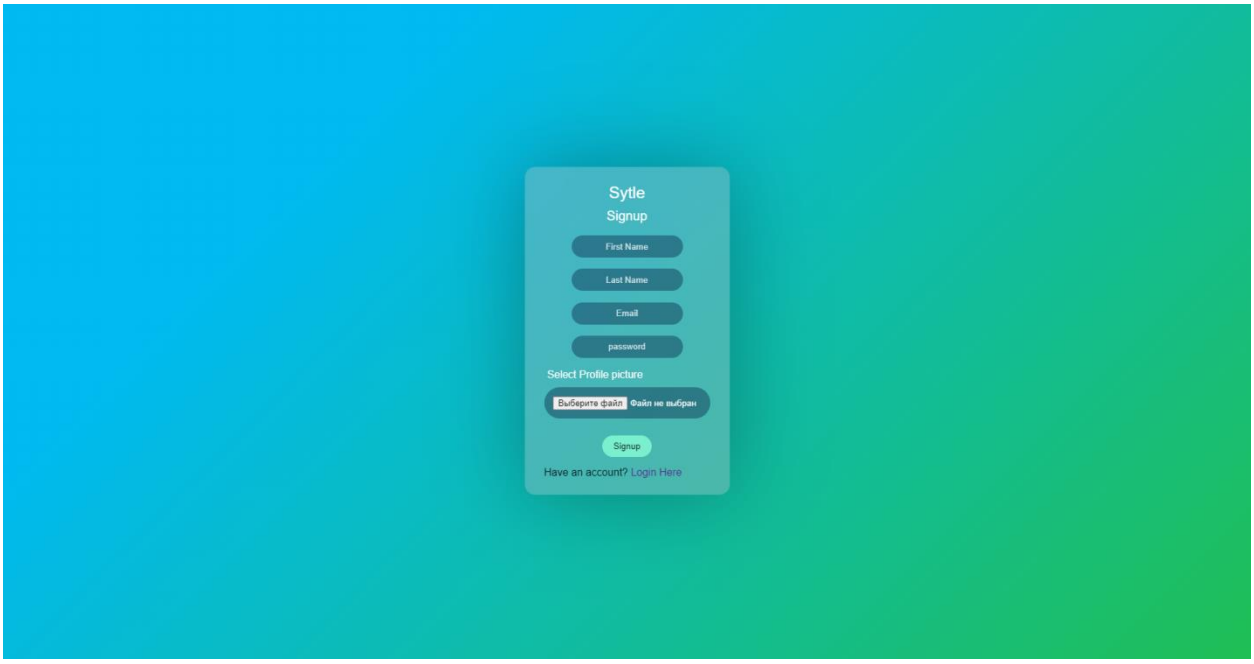


Рис. 14 Сторінка реєстрації користувача

1. Поле "Ім'я": В цьому полі користувач вводить своє ім'я.
2. Поле "Прізвище": Тут користувач вказує своє прізвище.

3. Поле "Електронна пошта": В даному полі користувач вводить свою електронну адресу, яка буде використовуватися для отримання сповіщень та комунікації.

4. Поле "Пароль": В цьому полі користувач обирає та вводить пароль для свого облікового запису. Пароль повинен відповідати вимогам безпеки і містити комбінацію символів.

5. Поле "Зображення": В цьому полі користувач може вставити своє зображення або обрати аватарку з доступних варіантів.

Після введення необхідної інформації користувачу надається можливість натиснути кнопку "Зареєструватися", щоб завершити процес реєстрації та створити свій обліковий запис. Також можуть бути присутні додаткові елементи і кнопки, які забезпечують можливість виконати інші дії, наприклад, переглянути умови використання, погодитися з політикою конфіденційності або перейти до інших сторінок, пов'язаних з реєстрацією користувача.

Сторінка реєстрації користувача має на меті забезпечити зручний і простий процес створення облікового запису, де користувач може надати необхідну особисту інформацію для входу до системи та подальшого використання ресурсів, функцій та послуг доступних на сайті.

Тепер давайте розглянемо перевірку коректності введеного електронного адресу (email). Перевірка коректного email є важливим етапом валідації даних, оскільки дозволяє перевірити, чи введений email відповідає прийнятим форматам і правилам.

Під час перевірки коректного email можуть застосовуватися різні правила, такі як перевірка наявності символу "@" у адресі, наявність доменної частини, валідація формату адреси тощо. Часто також використовуються готові бібліотеки або регулярні вирази для більш точної перевірки.

Наприклад, для перевірки коректного email можна використовувати наступний алгоритм:

1. Перевірка наявності символу "@" у введеному email. Цей символ є обов'язковим для коректного email.

2. Перевірка, що перед символом "@" присутній хоча б один символ.

3. Перевірка, що після символу "@" присутній хоча б один символ, і він відповідає формату домену (наприклад, .com, .net, .org тощо).

4. Перевірка формату email за допомогою регулярного виразу, який відповідає стандартним правилам для коректного email.

Це лише загальний приклад перевірки коректного email. В реальних проектах можуть використовуватися більш складні алгоритми валідації, залежно від конкретних вимог і правил, що встановлюються для введення email адреси. Нижче наведен приклад перевірки email, на її коректність(Рис.15). Та сам фрагмент коду перевірки(Рис.16)

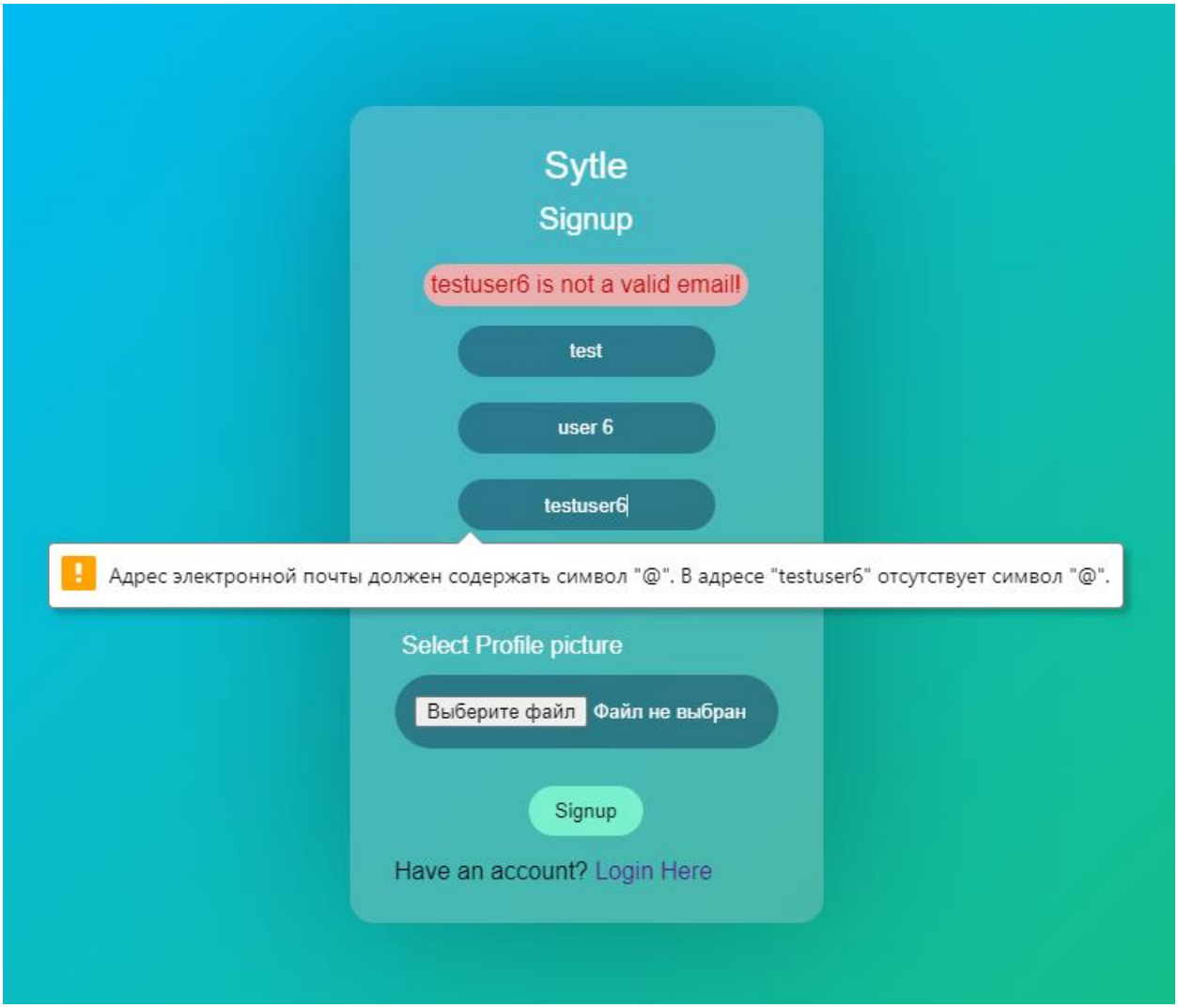


Рис. 15 Перевірка коректного email

```php
<?php
    session_start();
    include_once "config.php";
    $fname = mysqli_real_escape_string($conn, $_POST['fname']);
    $lname = mysqli_real_escape_string($conn, $_POST['lname']);
    $email = mysqli_real_escape_string($conn, $_POST['email']);
    $password = mysqli_real_escape_string($conn, $_POST['password']);
    $len=strlen($password);
    if(!empty($fname) && !empty($lname) && !empty($email) && !empty($password) && $len>=8){
        if(filter_var($email, filter: FILTER_VALIDATE_EMAIL)){
            $sql = mysqli_query($conn, query: "SELECT * FROM users WHERE email = '{$email}'");
            if(mysqli_num_rows($sql) > 0){
                echo "$email - This email already exist!";
            }else{
```

Рис. 16 Фрагмент коду перевірки email

Перевірка коректного зображення профілю полягає в тому, щоб переконатися, що вставлене зображення відповідає встановленим вимогам та належним чином обробляється системою. Після вставлення зображення користувачем, виконується набір перевірок для забезпечення його коректності.

У разі, якщо будь-яка з перевірок не пройдена успішно, в данному випадку на (Рис.17) зображення занадто велике, тому користувачу буде відображено повідомлення про помилку та вказано причину невдалої перевірки. В такому випадку користувачеві. А на (Рис.18) вказан сам фрагмент коду, який перевіряє зображення.

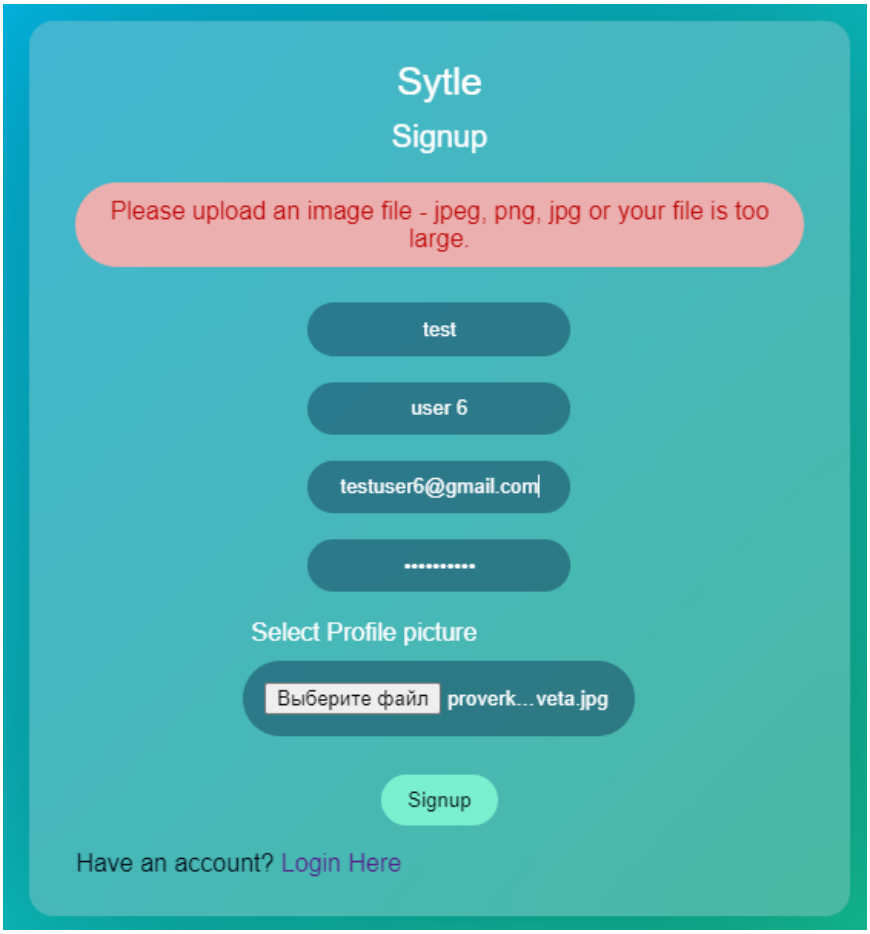


Рис. 17 Перевірка коректного зображення профілю

```
}else{
    if(isset($_FILES['image'])){
        $img_name = $_FILES['image']['name'];
        $img_type = $_FILES['image']['type'];
        $tmp_name = $_FILES['image']['tmp_name'];
        $img_explode = explode( separator: '.',$img_name);
        $img_ext = end( &array: $img_explode);
        $extensions = ["jpeg", "png", "jpg"];
        if(in_array($img_ext, $extensions) === true && $_FILES['image']['size'] < 1000000){
            $types = ["image/jpeg", "image/jpg", "image/png"];
            if(in_array($img_type, $types) === true){
                $time = time();
                $new_img_name = $time.$img_name;
                if(move_uploaded_file($tmp_name,"profileimage/".$new_img_name)){
                    $ran_id = rand(time(), 100000000);
                    $status = "Active now";
                    $encrypt_pass = password_hash($password, algo: PASSWORD_DEFAULT);
                    $insert_query = mysqli_query($conn, query: "INSERT INTO `users` (`user_id`,`unique_id`, `fname`, `lname`, `email`, `password`, `img`,`status`)
                    VALUES ( {$ran_id}','{$ran_id}', '{$fname}', '{$lname}', '{$email}', '{$encrypt_pass}', '{$new_img_name}', '{$status}' )");
                    if($insert_query){
                        $select_sql2 = mysqli_query($conn, query: "SELECT * FROM users WHERE email = '{$email}'");
                        if(mysqli_num_rows($select_sql2) > 0){
                            $result = mysqli_fetch_assoc($select_sql2);
                            $_SESSION['unique_id'] = $result['unique_id'];
                            echo "success";
```

Рис. 18 Фрагмент коду перевірки зображення

3.4 ВИСНОВОК ДО ТРЕТЬОГО РОЗДІЛУ

У розділі "Розробка проектних рішень" було проведено детальне вивчення та опис різних аспектів, пов'язаних з розробкою бази даних та веб-чату.

У розділі проведено розробку та створення таблиць для бази даних. Цей процес включав в себе аналіз вимог до даних, визначення сутностей та відносин між ними, а також створення структури таблиць з необхідними полями. Крім того, було розроблено процес створення таблиці в базі даних, який включав в себе кроки зі створення таблиці, визначення типів даних та обмежень.

Для забезпечення ефективної роботи бази даних була розроблена модель бази даних, яка включала в себе схему бази даних та відношення між таблицями. Ця модель допомогла уточнити структуру бази даних та забезпечити її оптимальну організацію.

Також був складений перелік таблиць, які були розроблені для проекту. Цей перелік включав назви таблиць, опис їхньої структури та зв'язків з іншими таблицями. Це сприяло кращому розумінню структури бази даних та полегшувало подальшу роботу з нею.

Архітектурний огляд веб-чату дозволив оцінити загальну структуру та організацію сайту. Було проаналізовано компоненти сайту, їхні взаємозв'язки та взаємодію з базою даних. Це дозволило забезпечити правильну інтеграцію функціональності сайту з базою даних та забезпечити його ефективну роботу.

Опис розробленого сайту та його інтерфейсу дав змогу детально представити функціональні можливості сайту, його структуру та інтерфейсні елементи. Було описано основні функції, які надає сайт, та способи взаємодії користувачів з ним. Це

дозволило зрозуміти, як сайт вирішує потреби користувачів та яким чином він забезпечує їхню зручність та задоволення.

В цілому, розділ "Розробка проектних рішень" розкриває процес розробки бази даних та веб-сайту, включаючи створення таблиць, проектування бази даних, архітектурний огляд веб-сайту та опис розробленого сайту.

ВИСНОВКИ

В ході роботи кваліфікаційної роботи було успішно реалізовано ряд завдань та досягнуто поставлених цілей.

Завдяки цьому, наш проект став привабливим та функціональним для користувачів. Розроблений веб-чат забезпечує зручний та інтуїтивно зрозумілий інтерфейс, що забезпечує зручну комунікацію та обмін повідомленнями.

У процесі розробки веб-чату ми використовували сучасні технології та інструменти, що дозволило нам досягти кращої продуктивності та ефективності. Також ми використовували сучасні мови програмування такі як JavaScript (jQuery, AJAX) та PHP, що дозволило нам забезпечити швидке та надійне функціонування веб-чату.

Продукт пройшов успішне тестування, під час якого було перевірено його функціональність, стабільність та безпеку. Крім того, документація, включаючи опис архітектури сайту, інструкції для користувачів та опис функціональності, була ретельно підготовлена. Це допоможе користувачам швидко ознайомитися з сайтом та зручно використовувати його.

В процесі роботи над дипломним проектом були здобуті наступні результати:

- Компоненти для розробки FrontEnd`у;
- Виконаний огляд популярних CSS фреймворків;
- Визначені головні переваги і функціонал фреймворку Bootsrap;
- Вибір засобу розробки для BackEnd`у;
- Вибір СУБД.;
- Вибір хостингу;
- Вибір системи управління версіями;

Загальним висновком є те, що розроблений проект веб-чат “Sytle” відповідає поставленим вимогам та має потенціал для подальшого розвитку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] Разработка со стороны Front end — что это такое и чем отличается от Back end: https://dan-it.com.ua/razrabotka-so-storony-front-end-chto-jeto-takoe-i-chem-otlichaetsja-ot-back-end/

[2]Що таке HTML та як за допомогою нього увійти в IT: https://mc.today/uk/shho-take-html-ta-yak-za-dopomogoyu-nogo-uvijti-do-it/

[3]Що таке CSS і для чого потрібен веб-розробнику: https://kiev.itstep.org/blog/what-is-css-and-why-does-a-web-developer-need-it

[4] JavaScript.: https://ru.wikipedia.org/wiki/JavaScript

[5]ТОП 10 лучших фреймворков для Front-end Dev: https://web-academy.com.ua/stati/336-10-front-end-dev-2018

[6]Верстать быстро и красиво: 15 популярных CSS фреймворков. URL: https://proglib.io/p/verstat-bystro-i-krasivo-15-populyarnyh-css-freymvorkov-2020-01-16

[7]Bootstrap: https://ru.wikipedia.org/wiki/Bootstrap_(%D1%84%D1%80%D0%B5%D0%B9%D0%BC%D0%B2%D0%BE%D1%80%D0%BA)

[8] Foundation: https://en.wikipedia.org/wiki/Foundation_(framework)

[9]Tailwind: https://channel9.msdn.com/Shows/Web-Wednesday/What-is-Tailwind-CSS#:~:text=Tailwind%20CSS%20is%20self%2Ddescribed,styles%20and%20change%20the%20layout.

[10]Система управління базами даних https://uk.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D1%96%D0

%BD%D0%BD%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85

[11]MySQL (СУБД)

https://uk.wikipedia.org/wiki/MySQL

[12] MongoDB (СУБД)

https://uk.wikipedia.org/wiki/MongoDB

[13]Який backend потрібний вашому веб-сайту?

https://goldwebsolutions.com/uk/blog/yakij-backend-potribnij-vashomu-veb-sajtu/

[14]XAMPP

https://www.apachefriends.org/ru/index.html

[15]GIT (система керування версіями)

https://uk.wikipedia.org/wiki/Git

ДОДАТОК А

Приклади коду програми

chat.js

```
    const form = document.querySelector(".typing-area"), inputfield = form.querySelector(".infield"), sendbtn = form.querySelector("button"), chat = document.querySelector(".chat-box");

    form.onsubmit = (e) => {
      e.preventDefault(); //preventing the default empty form ot be submitted
    }

    sendbtn.onclick = () => {

      let xhr = new XMLHttpRequest(); //creating XML object

      xhr.open("POST", "php/insert.php", true); //arguments are method,url,async
      xhr.onload = () => {
        if (xhr.readyState === XMLHttpRequest.DONE) {
          if (xhr.status === 200) {
            inputfield.value = ""; //once the msg is inserted into the DB then leave the input field blank
          }
        }
      }
      let formData = new FormData(form); //creating form data object
      xhr.send(formData);

    }
```

```
setInterval(() => {

   var oldscrollHeight = $(".chat-box")[0].scrollHeight - 20;

   let xhr = new XMLHttpRequest(); //creating XML object

   xhr.open("POST", "php/getmsg.php", true); //arguments are method,url,async
   xhr.onload = () => {
      if (xhr.readyState === XMLHttpRequest.DONE) {
         if (xhr.status === 200) {
            let data = xhr.response;
            console.log(data);
            chat.innerHTML = data;
            var newscrollHeight = $(".chat-box")[0].scrollHeight - 20;
            if (newscrollHeight > oldscrollHeight) {
               $(".chat-box").animate({
                  scrollTop: newscrollHeight
               }, 'normal'); //Autoscroll to bottom of div
            }
         }
      }
   }

   let formData = new FormData(form); //creating form data object
   xhr.send(formData);

}, 500);
```

login.js

```
    const form = document.querySelector(".login form"), continueBtn = form.querySelector(".signupbtn"), errortxt = form.querySelector(".error-txt");

    form.onsubmit = (e) => {
      e.preventDefault(); //preventing the default empty form ot be submitted
    }

    continueBtn.onclick = () => {
      let xhr = new XMLHttpRequest(); //creating XML object

      xhr.open("POST", "php/login.php", true); //arguments are method,url,async
      xhr.onload = () => {
        if (xhr.readyState === XMLHttpRequest.DONE) {
          if (xhr.status === 200) {
            let data = xhr.response;
            console.log(data);
            if (data === "success") {
              location.href = "../users.php"
            }
            else {
              errortxt.style.transition = "all 0.5s";
              errortxt.textContent = data;
              errortxt.style.background = "rgb(235, 175, 175)";
              errortxt.style.color = "rgb(192, 7, 7)";
              errortxt.style.display = "block";
            }
```

```
            }
        }
    }
    let formData = new FormData(form); //creating form data object
    xhr.send(formData);
}
```

signup.js

```
const form = document.querySelector(".signup form"), continueBtn = form.querySelector(".signupbtn"), errortxt = form.querySelector(".error-txt");

form.onsubmit = (e) => {
    e.preventDefault(); //preventing the default empty form ot be submitted
}

continueBtn.onclick = () => {
    let xhr = new XMLHttpRequest(); //creating XML object

    xhr.open("POST", "php/signup.php", true); //arguments are method,url,async
    xhr.onload = () => {
        if (xhr.readyState === XMLHttpRequest.DONE) {
            if (xhr.status === 200) {
                let data = xhr.response;
                console.log(data);
                if (data === "success") {
                    alert("Registration Successful")
                    location.href = "../users.php"
                }
```

```
            else {
                errortxt.style.transition = "all 0.5s";
                errortxt.textContent = data;
                errortxt.style.background = "rgb(235, 175, 175)";
                errortxt.style.color = "rgb(192, 7, 7)";
                errortxt.style.display = "block";
            }
        }
    }
    let formData = new FormData(form); //creating form data object
    xhr.send(formData);
}
```

users.js

```
const searchbar = document.querySelector(".users .search input"), searchbtn = document.querySelector(".users .search button"), userslist = document.querySelector(".users .users-list");


searchbar.onkeyup = () => {
    let searchterm = searchbar.value;

    //disabling the 500ms refresh on users list
    if (searchterm != "") {
        searchbar.classList.add("active");
    } else {
        searchbar.classList.remove("active");
```

```
    }


    let xhr = new XMLHttpRequest(); //creating XML object

    xhr.open("POST", "php/search.php", true); //arguments are method,url,async
    xhr.onload = () => {
      if (xhr.readyState === XMLHttpRequest.DONE) {
        if (xhr.status === 200) {
          let data = xhr.response;
          userslist.innerHTML = data;
        }
      }
    }
    xhr.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
    xhr.send("searchterm=" + searchterm);
  }


  setInterval(() => {
    let xhr = new XMLHttpRequest(); //creating XML object

    xhr.open("GET", "php/users.php", true); //arguments are method,url,async
    xhr.onload = () => {
      if (xhr.readyState === XMLHttpRequest.DONE) {
        if (xhr.status === 200) {
          let data = xhr.response;
```

```
            //if we are searching then there wont be refresh

            if (!searchbar.classList.contains("active")) {
                userslist.innerHTML = data;
            }
        }
    }
  }
  xhr.send();
}, 500);//this will run every 500ms
```

config.php

```
<?php

$server="localhost";
$username="danil";
$password = "danilpo31";
$database="chat";

$conn=mysqli_connect($server,$username,$password,$database);
if(!$conn){
   echo"<script>alert('Connection Failed.')</script>";
}



?>
```

getmsg.php

```
<?php
  session_start();
  if(isset($_SESSION['unique_id'])){
     include_once "config.php";
     //outgoing msg id
     $oid=$_SESSION['unique_id'];
     //incoming msg id
     $iid=mysqli_real_escape_string($conn,$_POST['incomingchat']);


     $dates="";

     $sql = mysqli_query($conn,"SELECT * FROM `message` WHERE (`iid` = {$iid} AND `oid` = {$oid}) OR (`iid` = {$oid} AND `oid` = {$iid}) ORDER BY msg_id");

     $output = "";

     if(mysqli_num_rows($sql)>0){
        while($row = mysqli_fetch_assoc($sql)){

           if($dates != $row['date']){
              $output .="<div class='dateincase'>".$row['date']."</div>";
              $dates=$row['date'];
           }

           if($row['oid'] === $oid){
```

```
                    //if this is send by the user
                    $output .='<div class="chat outgoing">
                                <div class="details">
                                    <p>'.$row['msg'].'<sub> '.$row['time'].'</sub></p>
                                </div>
                            </div>';
                }
                else{
                    //if this is send by the other side
                    $output .='<div class="chat incoming">
                                <div class="details">

<p>'.$row['msg'].'<sub> '.$row['time'].'</sub>'.'</p>
                                </div>
                            </div>';
                }
            }
        }else{
            $output .= '<div class="text">No messages are available. Once you send message they will appear here.</div>';
        }
            echo $output;
    }else{
        echo "<script> location.href='../login.php'; </script>";
    }


?>
```

insert.php

```
<?php
  session_start();
  if(isset($_SESSION['unique_id'])){
     date_default_timezone_set("Ukraine/Kiev");
     include_once "config.php";
     //outgoing msg id
     $oid=$_SESSION['unique_id'];
     //incoming msg id
     $iid=mysqli_real_escape_string($conn,$_POST['incomingchat']);
     //message
     $message = mysqli_real_escape_string($conn, $_POST['message']);
     $msg="";
     $dates=date("d/m/Y");

     $times=date("h:ia");
     $len=strlen($message);
     $point=0;

     while($point<$len){
        $msg=$msg.substr($message,$point,$point+50)." ";
        $point=$point+50;
     }

     if(!empty($msg)){
        $sql=mysqli_query($conn,"INSERT INTO `message` (`iid`, `oid` ,
`msg`,`date`,`time`) VALUES ('{$iid}','{$oid}','{$msg}','{$dates}','{$times}')") ;
```

```
        }
    }else{
        echo "<script> location.href='../login.php'; </script>";
    }


?>
```

list.php

```
<?php
    while($row = mysqli_fetch_assoc($query)){

        $sql2 = "SELECT * FROM `message` WHERE (`iid` = {$row['unique_id']} OR `oid` = {$row['unique_id']}) AND (`oid` = {$outgoing_id} OR `iid` = {$outgoing_id}) ORDER BY msg_id DESC LIMIT 1";

        $query2 = mysqli_query($conn, $sql2);

        $row2 = mysqli_fetch_assoc($query2);

        (mysqli_num_rows($query2) > 0) ? $result = $row2['msg'] : $result ="No message available";

        $cssprop="";

        ($row['status']=="offline")?$cssprop="offline":$cssprop="";

        (strlen($result)>28)?$msg=substr($result,0,28)."...":$msg=$result;
```

```
        $output .='<a href="chat.php?userid='.$row['unique_id'].'" class="rep">
                        <div class="content '.$cssprop.'" id="listing" >
                            <img          src="php/profileimage/'.$row['img'].'"          alt=""
width="75" heigh="75">
                        <div class="details">
                            <p><h4       style="margin-bottom:-10%;">'.$row['fname'].'
'.$row['lname'].'</h4><br>
                            '.$msg.'</p>

                        </div>
                    </div>
                </a>';
        }

    ?>
```

login.php

```
    <?php

    session_start();

    include_once "config.php";

    $email=mysqli_real_escape_string($conn,$_POST['email']);
    $password=mysqli_real_escape_string($conn,$_POST['password']);
```

```
    if(!empty($email) && !empty($password) ){

        $sql=mysqli_query($conn , "SELECT * FROM users WHERE email =
'{$email}'");

        if(mysqli_num_rows($sql)>0){

            $row=mysqli_fetch_assoc($sql);
            if($verify = password_verify($password, $row['password'])){
                $status="Active Now";
                $sql10 = mysqli_query($conn , "UPDATE users SET status = '{$status}'
WHERE unique_id = '{$row['unique_id']}'");

                if($sql10){
                $_SESSION['unique_id']=$row['unique_id'];
                echo "success";
                }
            }else{
                echo "Email or password is incorrect";
            }

        }else{
            echo "Email or password is incorrect";
        }

    }
    else{
        echo "All input fields are required !";
```

```
    }

    ?>
```

logout.php

```
    <?php

      session_start();
      if(isset($_SESSION['unique_id'])){
         include_once "config.php";
         $logoutid=mysqli_real_escape_string($conn , $_GET['logoutid']);

         if(isset($logoutid)){
         $status="offline";
         $sql = mysqli_query($conn , "UPDATE users SET status = '{$status}' WHERE unique_id = '{$_SESSION['unique_id']}'");

            if($sql){
               session_unset();
               session_destroy();
               echo "<script type='text/javascript'>location.href = '../login.php';</script>";
            }
         }else{
            echo "<script type='text/javascript'>location.href = '../users.php';</script>";
         }
```

```
    }else{
        echo "<script type='text/javascript'>location.href = '../login.php';</script>";
    }


?>
```

search.php

```
<?php
    session_start();
    $outgoing_id=$_SESSION['unique_id'];
    include_once "config.php";
    $searchTerm = mysqli_real_escape_string($conn, $_POST['searchterm']);
    $outgoing_id = $_SESSION['unique_id'];
    $output="";
    $sql = "SELECT * FROM users WHERE NOT unique_id = {$outgoing_id} AND
(fname LIKE '%{$searchTerm}%' OR lname LIKE '%{$searchTerm}%') ";
    $query = mysqli_query($conn, $sql);
    if(mysqli_num_rows($query)>0){
        include "list.php";
    }else{
        $output .="No user found !";
    }
    echo $output;

?>
```

signup.php

```php
<?php
    session_start();
    include_once "config.php";
    $fname = mysqli_real_escape_string($conn, $_POST['fname']);
    $lname = mysqli_real_escape_string($conn, $_POST['lname']);
    $email = mysqli_real_escape_string($conn, $_POST['email']);
    $password = mysqli_real_escape_string($conn, $_POST['password']);
    $len=strlen($password);
    if(!empty($fname) && !empty($lname) && !empty($email) && !empty($password) && $len>=8){
        if(filter_var($email, FILTER_VALIDATE_EMAIL)){
            $sql = mysqli_query($conn, "SELECT * FROM users WHERE email = '{$email}'");
            if(mysqli_num_rows($sql) > 0){
                echo "$email - This email already exist!";
            }else{
                if(isset($_FILES['image'])){
                    $img_name = $_FILES['image']['name'];
                    $img_type = $_FILES['image']['type'];
                    $tmp_name = $_FILES['image']['tmp_name'];
                    $img_explode = explode('.',$img_name);
                    $img_ext = end($img_explode);
                    $extensions = ["jpeg", "png", "jpg"];
                    if(in_array($img_ext, $extensions) === true && $_FILES['image']['size'] < 1000000){
                        $types = ["image/jpeg", "image/jpg", "image/png"];
                        if(in_array($img_type, $types) === true){
```

```
                    $time = time();
                    $new_img_name = $time.$img_name;

if(move_uploaded_file($tmp_name,"profileimage/".$new_img_name)){
                        $ran_id = rand(time(), 100000000);
                        $status = "Active now";
                        $encrypt_pass                                                    =
password_hash($password,PASSWORD_DEFAULT);
                        $insert_query = mysqli_query($conn, "INSERT INTO `users`
(`user_id`,`unique_id`, `fname`, `lname`, `email`, `password`, `img`,`status`)
                        VALUES      ('{$ran_id}','{$ran_id}',      '{$fname}','{$lname}',
'{$email}', '{$encrypt_pass}', '{$new_img_name}', '{$status}')");
                        if($insert_query){
                            $select_sql2 = mysqli_query($conn, "SELECT * FROM users
WHERE email = '{$email}'");
                            if(mysqli_num_rows($select_sql2) > 0){
                                $result = mysqli_fetch_assoc($select_sql2);
                                $_SESSION['unique_id'] = $result['unique_id'];
                                echo "success";
                            }else{
                                echo "This email address not Exist!";
                            }
                        }else{
                            echo "Something went wrong. Please try again!";
                        }
                    }
                }else{
                    echo "Please upload an image file - jpeg, png, jpg";
```

```
                    }
                }else{
                    echo "Please upload an image file - jpeg, png, jpg or your file is too large.";
                }
            }
        }
    }else{
        echo "$email is not a valid email!";
    }
}else{
    echo "All input fields are required!";
}
?>
```

users.php

```
<?php
  session_start();
  include_once "config.php";
  $outgoing_id = $_SESSION['unique_id'];
  $sql = "SELECT * FROM users WHERE NOT unique_id = {$outgoing_id}";
  $query = mysqli_query($conn, $sql);
  $output = "";
  if(mysqli_num_rows($query) == 0){
    $output .= "No users are available to chat";
  }elseif(mysqli_num_rows($query) > 0){
    include_once "list.php";
  }
```

```
    echo $output;
?>
```

chat.php

```
<?php
session_start();
include_once "php/config.php";
if(!isset($_SESSION['unique_id'])){
    echo "<script> location.href='login.php'; </script>";
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- <meta http-equiv="refresh" content="3"> -->
  <link rel="stylesheet" href="style2.css">
  <title>Chat Application</title>
</head>
<body>
  <div class="wrapper">
      <section class="chat-area">
        <header>

          <?php
          include_once "php/config.php";
```

```
                $userid = mysqli_real_escape_string($conn, $_GET['userid']);
                $sql = mysqli_query($conn, "SELECT * FROM users WHERE unique_id
= {$userid}");
                if(mysqli_num_rows($sql) > 0){
                    $row=mysqli_fetch_assoc($sql);
                    $cssprop="";
                    $imgsrc="";
                    ($row['status']=="offline")?$cssprop="offline":$cssprop="";

($row['status']=="offline")?$imgsrc="pics/grey.png":$imgsrc="pics/greencircle.png
";

                }else{
                    echo "<script> location.href='.php'; </script>";
                }

                ?>

                <div class="content">

                    <div class="details">

                        <span>
                            <a href="users.php">
                                <img                src="pics/arrow.png"                alt=""
style="transform:rotate(180deg);">
                            </a>
```

```
                    <img src="php/profileimage/<?php echo $row['img']?>" alt="" width="50" class="profilepic">
                    <div class="<?php echo $cssprop?>">
                        <?php echo $row['fname']." ".$row['lname'] ?><br>
                        <p>
                            <img src="<?php echo $imgsrc?>" alt="" width="15" height="15px" style="transform: translateY(2.2px);"><?php echo $row['status'] ?>
                        </p>
                    </div>
                </span>
            </div>
        </div>
    </header>
    <div class="chat-box">

    </div>
    <form action="" class="typing-area">
        <input type="text" name="outgoingchat" value="<?php echo $_SESSION['unique_id']; ?>" style="display:none;">
        <input type="text" name="incomingchat" value="<?php echo $userid;?>" style="display:none;">
        <input type="text" placeholder="Type a Message...." class="infield" name="message" autocomplete="off">
        <button class="msgbtn"><img src="pics/send.png" alt="" width="32" style="transform:rotate(90deg)"></button>
    </form>
</section>
</div>
```

```
    <script                                                    type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script src="javascript/chat.js"></script>
  </body>
  </html>
```

import.sql

```
create DATABASE IF NOT EXISTS chat CHARSET utf8mb4;
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";
use chat;

CREATE TABLE `message` (
 `msg_id` int(11) NOT NULL,
 `iid` int(255) NOT NULL,
 `oid` int(255) NOT NULL,
 `msg` varchar(1000) NOT NULL,
 `date` varchar(40) NOT NULL,
 `time` varchar(40) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

ALTER TABLE `message`
 ADD PRIMARY KEY (`msg_id`);

ALTER TABLE `message`
 MODIFY `msg_id` int(11) NOT NULL AUTO_INCREMENT;
```

```
COMMIT;

CREATE TABLE `users` (
                    `user_id` int(11) NOT NULL,
                    `unique_id` int(200) NOT NULL,
                    `fname` varchar(255) NOT NULL,
                    `lname` varchar(255) NOT NULL,
                    `email` varchar(255) NOT NULL,
                    `password` varchar(255) NOT NULL,
                    `img` varchar(400) NOT NULL,
                    `status` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

index.php

```
<?php
 session_start();
 if(isset($_SESSION['unique_id'])){
   echo "<script> location.href='users.php'; </script>";
 }
?>
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- <meta http-equiv="refresh" content="3"> -->
  <link rel="stylesheet" href="style.css">
```

```
    <title>Chat Application</title>
  </head>
  <body>
    <div class="wrapper">
      <section class="form signup">
        <header>Sytle</header>
        <h2 style="color:white;text-align:center;font-weight:200;font-size:20px">Signup</h2>
        <br>
        <form action="#" enctype="multipart/form-data">
          <div class="error-txt">
            This is an error message!
          </div>
          <div class="name-details">
            <div class="field">
              <input type="text" placeholder="First Name" name="fname" autocomplete="off" required>
            </div>
            <div class="field">
              <input type="text" placeholder="Last Name" name="lname" autocomplete="off" required>
            </div>
          </div>
          <div class="field">
            <input type="email" placeholder="Email" name="email" autocomplete="off" required>
          </div>
          <div class="field">
```

```
                    <input type="password" placeholder="password" name="password" autocomplete="off"  minlength="8" required>
                </div>
                <div class="field" id="profile">
                    <label>Select Profile picture</label>
                    <input type="file" name="image" required>
                </div>
                <div class="field" id="signupbutton">
                    <input type="submit" value="Signup" class="signupbtn">
                </div>
            </form>
            <div class="link">Have an account? <a href="login.php">Login Here</a></div>
        </section>
    </div>

    <script src="javascript/signup.js"></script>
</body>
</html>
```

login.php

```
<?php
 session_start();
 if(isset($_SESSION['unique_id'])){
   echo "<script> location.href='users.php'; </script>";
 }
?>
<!DOCTYPE html>
```

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- <meta http-equiv="refresh" content="3"> -->
  <link rel="stylesheet" href="style.css">
  <title>Chat Application</title>
</head>
<body>
  <div class="wrapper">
    <section class="form login">
      <header>Sytle</header>
      <h2 style="color:white;text-align:center;font-weight:200;font-size:20px">Login</h2>
      <br>
      <form action="#">
        <div class="error-txt">
        </div>
        <div class="field">
          <input type="email" placeholder="Email" name="email" autocomplete="off">
        </div>
        <div class="field">
          <input type="password" placeholder="Password" name="password" autocomplete="off" minlength="8">
        </div>
          <div class="field" id="signupbutton">
```

```
                    <input type="submit" value="Login" class="signupbtn">
                </div>
            </form>
            <div class="link" style="font-size:14px">Don't have an account? <a href="index.php">Signup Here</a></div>
        </section>
    </div>
    <script src="javascript/login.js"></script>
</body>
</html>
```

style.css

```
* {
 margin: 0;
 padding: 0;
 box-sizing: border-box;
 text-decoration: none;
 font-family: "Poppins", sans-serif;
}
body {
 display: flex;
 align-items: center;
 justify-content: center;
 min-height: 100vh;
 background-color: #20bf55;
 background-image: linear-gradient(315deg, #20bf55 0%, #01baef 74%);
}
.wrapper {
```

```
 background-color: #a2b1b665;
 backdrop-filter: blur(25px);
 transition: all 0.5s;
 border-radius: 16px;
 box-shadow: 0 0 128px 0 rgba(0, 0, 0, 0.3),
   0 32px 64px -48pc rgba(0, 0, 0, 0.5);
}

.form {
 padding: 25px 30px;
 color: #010c16;
}

form {
 display: flex;
 flex-direction: column;
 justify-content: center;
 align-items: center;
 color: white;
}

.form header {
 font-size: 25px;
 font-weight: 500;
 padding-bottom: 10px;
 color: rgb(255, 255, 255);
 text-align: center;
}
```

```css
.form input {
 margin-bottom: 10%;
 width: 100%;
 transition: all 0.5s;
 display: flex;
 justify-items: center;
 align-items: center;
 padding: 5%;

 border-radius: 50px;
 background-color: rgba(11, 48, 80, 0.459);
 border: 1px solid rgba(255, 255, 255, 0);
 /* backdrop-filter: blur(50px); */
 text-align: center;
 color: white;
}

::placeholder {
 color: white;
 opacity: 0.8;
}

.form input:hover {
 transform: translateY(-8%);
 transition: all 0.5s;
}

input.signupbtn {
```

```
 padding: 8px 16px;
 display: flex;
 justify-content: center;
 background-color: rgba(127, 255, 212, 0.795);
 backdrop-filter: blur(25px);
 color: #010c16;
 font-weight: 500;
 margin-bottom: 15px;
 cursor: pointer;
}

div#signupbutton {
 cursor: pointer;
 text-align: center;
}

div#profile label {
 margin-left: 2%;
}

div#profile input {
 margin-top: 10px;
 display: flex;
 justify-content: center;
 align-items: center;
}

div.link a:hover {
```

```
 color: white;
 transition: all 0.3s;
}

input:focus,
textarea:focus,
select:focus {
 outline: none;
}

.form form .error-txt {
 transition: all 0.5s;
 text-align: center;
 color: rgb(192, 7, 7);
 font-weight: 300;
 background-color: rgb(235, 175, 175);
 border-radius: 30px;
 padding: 2%;
 display: none;
 margin-bottom: 5%;
}
h3 {
 text-align: center;
 color: white;
}
```

style2.css

```
* {
```

```
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  text-decoration: none;
  font-family: "Poppins", sans-serif;
}
body {
  display: flex;
  align-items: center;
  justify-content: center;
  min-height: 100vh;
  background-color: #20bf55;
  background-image: linear-gradient(315deg, #20bf55 0%, #01baef 74%);
}
.wrapper {
  min-width: 50%;
  min-height: 50vh;
  background-color: #a2b1b665;
  backdrop-filter: blur(25px);
  transition: all 0.5s;
  border-radius: 16px;
  box-shadow: 0 0 128px 0 rgba(0, 0, 0, 0.3),
    0 32px 64px -48pc rgba(0, 0, 0, 0.5);
}
@media (max-width: 420px) {
  .wrapper {
    min-width: 90%;
  }
```

```
}

::placeholder {
 color: white;
 opacity: 0.5;
}

div.link a:hover {
 color: white;
 transition: all 0.3s;
}

div.search input {
 padding: 1%;
 width: 100%;
 border-radius: 50px;
 background-color: rgba(11, 48, 80, 0.459);
 border: 1px solid rgba(255, 255, 255, 0);
 backdrop-filter: blur(50px);
 text-align: center;
 color: white;
}
section.users {
 margin: 5%;
}
button#searchbutton {
 cursor: pointer;
 transition: all 0.5s;
```

```
 margin-left: 2%;
 /* transform: translateY(2px); */
 padding: 1%;
 border-radius: 50px;
 background-color: rgba(11, 48, 80, 0);
 border: 1px solid rgba(255, 255, 255, 0);
 text-align: center;
 color: white;
}

button#searchbutton:hover {
 transform: scale(1.2);
 transition: all 0.5s;
}

div p {
 color: rgb(6, 160, 6);
}
div.offline p {
 color: rgb(75, 75, 75);
}
.users header {
 display: flex;
 align-items: flex-start;
 padding-bottom: 5px;
 justify-content: space-between;
 border-bottom: 3px solid rgba(114, 114, 114, 0.315);
}
```

```css
div.details span {
 margin-bottom: 5px;
 width: 120%;
 display: flex;
 justify-content: space-evenly;
 align-items: center;
 font-size: 15px;
}
section.users header a {
 transition: all 0.5s;
 display: flex;
 justify-content: center;
 background-color: rgba(127, 255, 212, 0.479);
 backdrop-filter: blur(25px);
 color: #010c16;
 font-weight: 500;
 border-radius: 10px;
 padding: 1%;
}
section.users header a:hover {
 transform: translateY(-5%);
 transition: all 0.5s;
}

div.search {
 margin-top: 2%;
 display: flex;
 justify-content: space-between;
```

```
 align-items: center;
}
div.search div {
 display: flex;
 justify-content: right;
 align-items: center;
}

input:focus,
textarea:focus,
select:focus {
 outline: none;
}

div#lists img {
 border-radius: 50%;
 width: 75px;
 height: 75px;
}
div#lists div.content {
 background-color: rgba(153, 252, 158, 0.322);
 backdrop-filter: blur(50px);
 padding: 1.5%;
 border-radius: 35px;
 text-align: left;
 margin-top: 5%;
 display: flex;
 justify-content: left;
```

```
 align-items: center;
 color: rgb(0, 0, 0);
 font-weight: 300;
}

div#lists div.content.offline {
 background-color: rgba(42, 43, 42, 0.322);
}

div#lists a div.details p {
 font-weight: 300;
 color: rgb(0, 0, 0);
}

div#lists div.details {
 margin-left: 5%;
}

div#lists {
 max-height: 350px;
 overflow-x: hidden;
 overflow-y: auto;
}

div#lists::-webkit-scrollbar {
 width: 0px;
 visibility: hidden;
}
```

```css
.users #listing .details {
 text-align: left;
}

/* chat page css */
div.chat-box div.chat p {
 padding: 8px 16px;
}

.content .details span img.profilepic {
 border-radius: 50%;
 width: 75px;
 height: 75px;
}

.chat-box {
 height: 60vh;
 overflow-x: hidden;
 overflow-y: auto;
}
.chat-box::-webkit-scrollbar {
 width: 0;
 visibility: hidden;
}

.chat-box .outgoing,
.chat-box .incoming {
```

```
 font-weight: 300;
 display: flex;
}

.outgoing .details {
 border-radius: 18px 18px 0 18px;
 background: rgba(8, 160, 58, 0.541);
 padding-right: 20%;
 display: flex;
 margin: 3%;
 margin-top: 1%;
 padding-right: 1%;
 margin-left: auto;
 max-width: 60%;
}

.incoming .details {
 margin: 3%;
 margin-top: 1%;
 border-radius: 0 18px 18px 18px;
 background-color: #404c5096;
 padding-left: 1%;
 margin-right: auto;
 max-width: 75%;
}

.incoming .details p {
 text-align: justify;
```

```
 color: white;
}

.outgoing .details p {
 text-align: justify;
 text-justify: auto;
 color: white;
}

section.chat-area header div.content {
 background-color: #ffffff41;
 border-radius: 16px 16px 0 0;
}

section.chat-area header div.content div.details {
 display: inline-block;
 transition: all 0.5s;
 margin: 3%;
}

section.chat-area header a {
 transition: all 0.5s;
}

section.chat-area header a:hover {
 transform: translateX(-10%);
 transition: all 0.5s;
}
```

```css
section.chat-area .typing-area input {
 transition: all 0.5s;
 min-width: 90%;
 padding: 1%;
 margin: 1.5% 1%;
 border-radius: 50px;
 background-color: rgba(11, 48, 80, 0.459);
 border: 1px solid rgba(255, 255, 255, 0);
 backdrop-filter: blur(50px);
 text-align: center;
 color: white;
}

section.chat-area .typing-area button {
 cursor: pointer;
 transition: all 0.5s;
 background-color: rgba(11, 48, 80, 0);
 border: 0px solid rgba(255, 255, 255, 0);
}

section.chat-area form {
 display: flex;
 justify-content: center;
 align-items: center;
}

section.chat-area form button:hover {
```

```
 transform: translatex(10%);
 transition: all 0.5s;
}

div.dateincase {
 display: flex;
 justify-content: center;
 align-items: center;
 margin-left: 45%;
 margin-top: 5%;
 margin-bottom: 5%;
 border-radius: 10px;
 padding: 5px;
 width: min-content;
 background-color: #404c5096;
 color: rgb(255, 255, 255);
 opacity: 0.7;
}

sub {
 font-size: 10px;
}
```

user.php

```
<?php

session_start();
include_once "php/config.php";
```

```php
    if(!isset($_SESSION['unique_id'])){
        echo "<script> location.href='login.php'; </script>";
    }
    ?>

    <!DOCTYPE html>
    <html lang="en">
    <head>
      <meta charset="UTF-8">
      <meta http-equiv="X-UA-Compatible" content="IE=edge">
      <meta name="viewport" content="width=device-width, initial-scale=1.0">
      <!-- <meta http-equiv="refresh" content="3"> -->
      <link rel="stylesheet" href="style2.css">
      <title>Chat Application</title>
    </head>
    <body>
      <div class="wrapper">
        <section class="users">
          <header>

            <?php
              include_once "php/config.php";
               $sql = mysqli_query($conn, "SELECT * FROM users WHERE unique_id = {$_SESSION['unique_id']}");
              if(mysqli_num_rows($sql) > 0){
                $row = mysqli_fetch_assoc($sql);
              }
            ?>
```

```
            <div class="content">

                <div class="details">

                    <span>
                        <img src="php/profileimage/<?php echo $row['img'];?>" alt="Something went Wrong" width="60" height="60" class="profilepic">
                        <div>
                            <?php echo $row['fname']." ".$row['lname'] ?><br>
                            <p>
                                <img src="pics/greencircle.png" alt="" width="15" height="15px" style="transform: translateY(2.2px);"> <?php echo $row['status']; ?>
                            </p>
                        </div>
                    </span>


                </div>
            </div>
            <a href="php/logout.php?logoutid=<?php echo $_SESSION['unique_id']?>" class="logout" style="margin-left:5%;">Logout</a>
        </header>
        <div class="search">
            <span class="text">Search a user</span>
            <div>
            <input type="text" placeholder="Enter email or username">
```

```
            </div>
        </div>

        <div class="users-list" id="lists">

        </div>

    </section>
  </div>

  <script src="javascript/users.js"></script>
</body>
</html>
```