Real Time Chat App

автор: Кристиян Димитров

специалност, курс: Софтуерни технологии и Дизайн, 2 курс

фн: 2301681013

**Съдържание**:

**[Увод](#_Toc193882502)** [1](#_Toc193882502)

[**Определяне на целите** 1](#_Toc193882503)

[**Предметна област** 1](#_Toc193882504)

[**Приложение в различни сфери на използване** 1](#_Toc193882505)

[**Основни функционалности** 1](#_Toc193882506)

[**Архитектура на системата** 1](#_Toc193882507)

[**Основни технологии** 1](#_Toc193882508)

[**Бази данни** 1](#_Toc193882509)

[**Проектиране и структура** 1](#_Toc193882510)

[**Регистрация** 1](#_Toc193882511)

[**Потребителски интерфейс** 1](#_Toc193882512)

[**Редактирране на профил** 1](#_Toc193882513)

[**Chat страница** 1](#_Toc193882514)

[**CSS** 1](#_Toc193882515)

[**Основен фон и стил на тялото(body)** 1](#_Toc193882516)

[**Прозрачен и модерен дизайн за формата** 1](#_Toc193882517)

[**Текстови полета с ефект при фокусиране** 1](#_Toc193882518)

[**Съобщения в чат (изпратени и получени)** 1](#_Toc193882519)

[**Типично поле за въвеждане с бутон за изпращане** 1](#_Toc193882520)

[**Стил за списъка с потребители** 1](#_Toc193882521)

[**Адаптивен дизайн за мобилни устройства** 1](#_Toc193882522)

[**Заключение** 1](#_Toc193882523)

# **Увод**

Проектът представлява уеб-базирано чат приложение, което позволява на потребителите да комуникират помежду си в реално време. Той включва функционалности като регистрация, влизане в системата, търсене на потребители и изпращане на съобщения. Разработката е насочена към създаване на сигурна и удобна среда за комуникация чрез използване на PHP, MySQL, JavaScript и AJAX.

# **Определяне на целите**

Целта на този проект е да предостави лесен и удобен начин за комуникация между потребители чрез минималистичен и интуитивен интерфейс. Той може да бъде използван за лични разговори, работа в екип или социални мрежи. Освен това, проектът има за цел:

* **Осигурява бърза и надеждна комуникация.** Приложението трябва да поддържа мигновена доставка на съобщения, без забавяния, при условие, че потребителите имат валидна интернет връзка. Това включва използване на технологии за осъществяване на реалновременна комуникация, като ускорени протоколи, WebSockets или AJAX заявки.
* **Предлага интуитивен и модерен потребителски интерфейс.** Интерфейсът е проектиран с внимание към детайла, като модерен дизайн, минимализъм и лесна навигация са основни принципи, които спомагат за по-доброто потребителско изживяване.
* **Гарантира сигурност и защита на данните.** В съвременните комуникационни платформи е от съществено значение да се гарантира защита на личните данни. Проектът включва внедряване на силни криптографски методи, контрол на достъпа и мерки за предотвратяване на неоторизиран достъп.
* **Поддържа мултимедия и обмен на файлове.** Освен текстова комуникация, приложението трябва да предоставя възможност за споделяне на изображения, видеа и други формати на файлове. Това улеснява по-богата и разнообразна комуникация помежду на потребителите.
* **Обезпечава мащабируемост и бъдещо развитие.** Архитектурата на приложението е изградена по модулен принцип, което позволява лесно разширяване и добавяне на нови функционалности или интеграции с други системи в бъдеще.

# **Предметна област**

Чат приложението се позиционира като ценен инструмент както за индивидуални потребители, така и за бизнес клиенти. В контекста на нарастващото значение на дигиталната комуникация, особено в ситуации, където физическите срещи са ограничени, този проект цели да отговори на нуждите от бърза и надеждна комуникация. Някои от ключовите аспекти, които подчертават значимостта на проекта, са:

* **Подобряване на екипната комуникация в работни среди.** Много компании и екипи имат нужда от вътрешна платформа за моментално съобщаване, която да улесни сътрудничеството и обмена на информация в реално време.
* **Улесняване на социалната интеракция.** За потребители, които искат незабавно да обменят лични съобщения или да се срещнат онлайн, приложението предлага възможност за изграждане на общности, групови разговори и лесен достъп до личния чат.
* **Интеграция със съществуващи бизнес процеси.** Чрез лесната си конфигурация и мащабируемост, приложението може да бъде интегрирано с други корпоративни решения или CRM системи, осигурявайки единна платформа за комуникация и управление на данни.
* **Реакция на нарастващите изисквания на технологичната среда.** С разрастването на интернет приложенията и увеличаването на броя на потребителите, е необходимо създаването на приложения, които да могат да посрещнат високия трафик и действително да осигурят безпроблемна работа дори и при натоварени сървъри.

# **Приложение в различни сфери на използване**

Чат приложението има потенциал да бъде използвано не само в комуникационните мрежи, но и в по-специфични области като:

* **Корпоративни комуникации.** В големи предприятия и организации, където времето за реакция е от решаващо значение, системата подпомага вътрешната комуникация, позволявайки бърз обмен на информация, сътрудничество и координация между отделните звена.
* **Образователни платформи.** Чрез интеграция на функционалности като групови чатове и видео-конференции, приложението може да бъде адаптирано за учебни среди, където студентите и преподавателите имат нужда от бърза и интерактивна комуникация.
* **Социални мрежи и общности.** Благодарение на поддръжката на мултимедия, групови дискусии и персонални профили, платформата може да послужи като основа за създаване на социални мрежи, където потребителите могат активно да обменят идеи, снимки и мнения.

# **Основни функционалности**

* **Регистрация на потребители**: Потребителите могат да се регистрират чрез предоставяне на име, фамилия, имейл, парола и профилна снимка.Лични съобщения в реално време между потребители.
* **Вход за потребители**: Потребителите могат да влизат в приложението с имейл и парола.
* **Страница с потребителски профили и статуси**: Потребителите могат да видят информация за своето потребителско име, фамилия, статус и профилна снимка.
* **Търсене на потребители**: Потребителите могат да търсят други потребители по име.
* **Чат в реално време**: Потребителите могат да изпращат съобщения в реално време след успешен вход.
* **Отзивчив дизайн**, подходящ за мобилни устройства
* **Сигурност** чрез хеширане на пароли и сесии

# **Архитектура на системата**

Проектът използва трислойна архитектура:

1. **Клиентски слой** – HTML, CSS и JavaScript за потребителския интерфейс.
2. **Сървърен слой** – PHP за обработка на заявки и комуникация с базата данни.
3. **База данни** – MySQL за съхранение на потребителски данни и съобщения.

# **Основни технологии**

* **Frontend**: HTML, CSS, JavaScript.
* **Backend**: PHP за обработка на входни данни, регистрация, вход и чат съобщения.
* **Database**: MySQL за съхранение на потребителски данни и съобщения

# **Бази данни**

Таблица users

|  |  |  |
| --- | --- | --- |
| **Колона** | **Тип** | **Описание** |
| id | INT (AUTO\_INCREMENT) | Уникален идентификатор |
| name | VARCHAR(255) | Потребителско име |
| email | VARCHAR(255) | Имейл адрес |
| password | VARCHAR(255) | Хеширана парола |
| status | ENUM('online', 'offline') | Статус на потребителя |

Таблица messages

|  |  |  |
| --- | --- | --- |
| Колона | Тип | Описание |
| id | INT (AUTO\_INCREMENT) | Уникален идентификатор |
| sender\_id | INT | Изпращач на съобщението |
| receiver\_id | INT | Получател на съобщението |
| message | TEXT | Съдържание на съобщението |
| timestamp | TIMESTAMP | Време на изпращане |

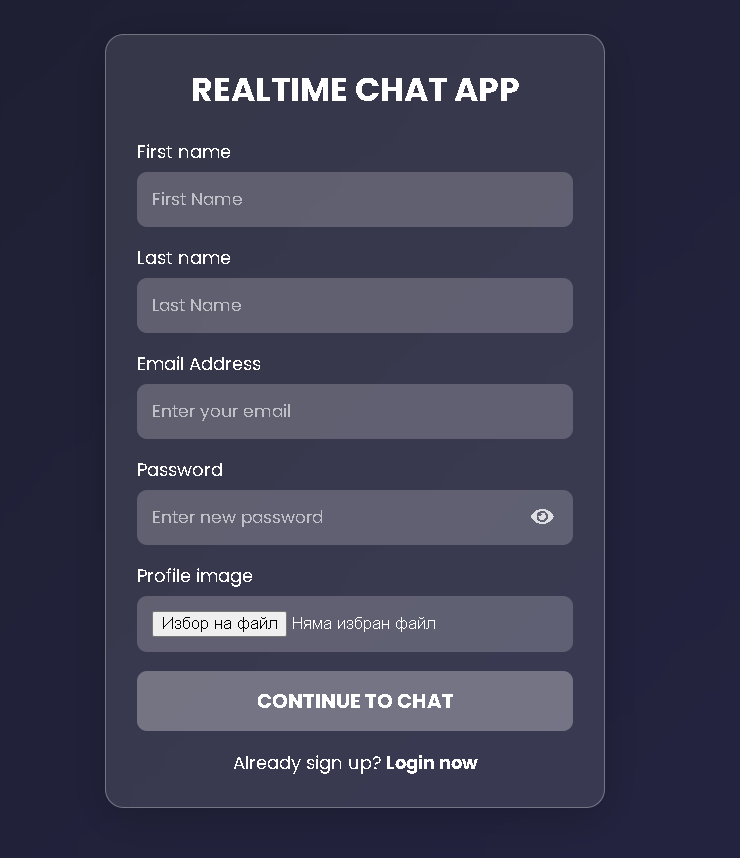
# **Проектиране и структура**

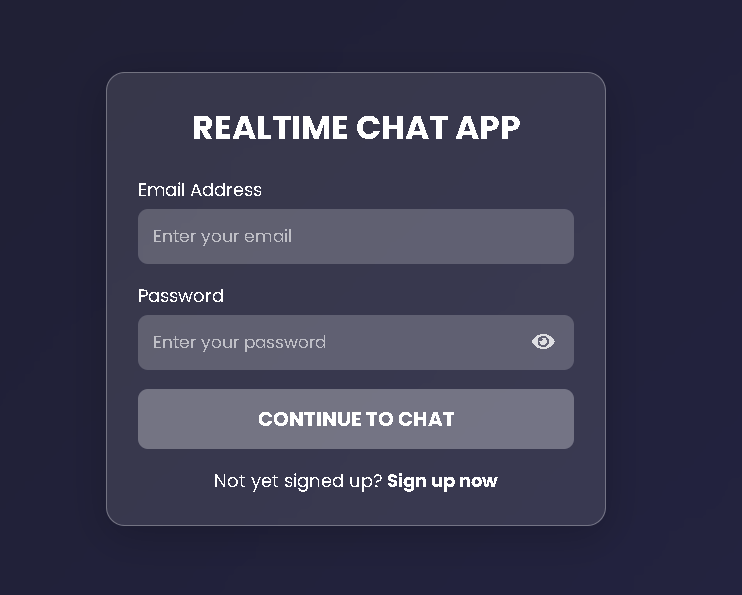
## **Регистрация**

Формата за регистрация позволява на нови потребители да създадат профил в уеб приложението.

Тя съдържа следните полета:

* **Име и фамилия** – Потребителят трябва да въведе своето име и фамилия.
* **Имейл адрес** – Използва се за идентификация и комуникация с потребителя.
* **Парола** – Задължително поле, което потребителят използва за защита на профила си.
* **Качване на профилна снимка** – Потребителят може да добави изображение, което ще се използва в чата.
* **Бутон за регистрация** – След попълване на всички полета, натискането на този бутон изпраща данните към сървъра за обработка.

След успешна регистрация потребителят бива пренасочен към страницата с наличните потребители, където може да започне чат.





Когато потребителят натисне бутона за изпращане, вместо формата да се изпрати по стандартния начин (и страницата да се презареди), използваме **AJAX** за асинхронно изпращане на данни към сървъра.

Сървърът (PHP скриптът signup.php) ще обработи тези данни (например, ще запише новия потребител в база данни) и ще върне отговор.

Ако отговорът от сървъра е "success", потребителят ще бъде пренасочен към друга страница (например, списък с потребители).

Ако има грешка (например, ако имейлът вече е използван), ще се покаже съобщение за грешка в елемента .error-text.



Този PHP скрипт обработва регистрацията на потребители в уеб приложение, като включва няколко ключови стъпки за валидиране, съхранение и защита на данните.

Първо, скриптът стартира сесия (session\_start()), което позволява запазване на потребителска информация през различни страници. След това се включва конфигурационен файл (config.php), който съдържа връзката с базата данни. За улесняване на отстраняването на грешки се активира детайлно докладване на грешки с error\_reporting(E\_ALL) и ini\_set('display\_errors', 1), което помага при разработката и откриването на проблеми.

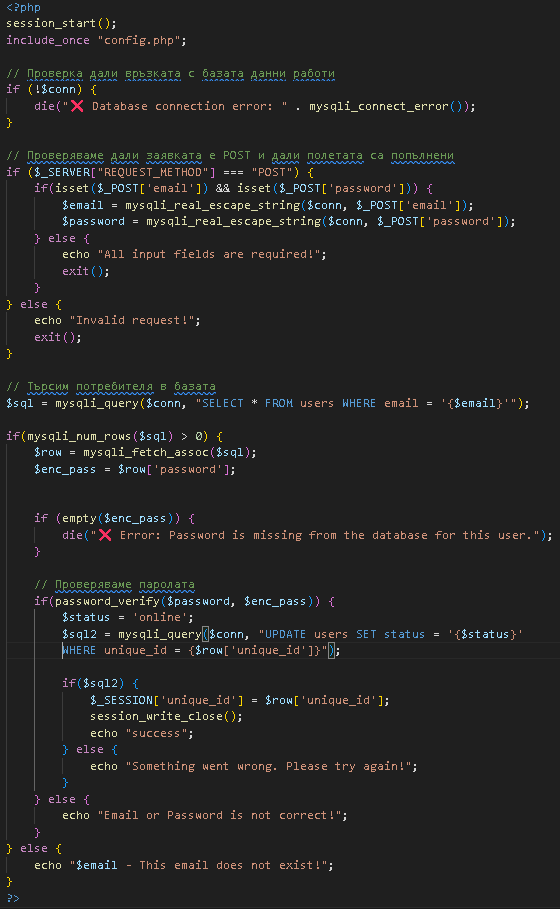
Следващата важна стъпка е обработката на въведените от потребителя данни. Скриптът приема данните от формуляра чрез $\_POST и ги обработва с mysqli\_real\_escape\_string(), за да се избегнат SQL инжекции. След това се проверява дали всички задължителни полета (име, фамилия, имейл и парола) са попълнени. Ако някое от тях липсва, се връща съобщение за грешка.

Освен това, преди да продължи със създаването на нов акаунт, скриптът валидира въведения имейл с filter\_var($email, FILTER\_VALIDATE\_EMAIL). Ако имейлът е невалиден, потребителят получава съобщение за грешка. Следва допълнителна проверка дали въведеният имейл вече съществува в базата данни. Ако той вече е регистриран, регистрацията не се осъществява.

За допълнителна функционалност е добавена възможност за качване на профилна снимка. Скриптът проверява дали е качен файл и дали форматът на файла е сред позволените (jpeg, png, jpg). Ако всичко е наред, изображението се запазва в папка images/ със специално генерирано име, базирано на текущото време, за да се избегнат дублирания.

След като всички проверки са преминати успешно, се създава уникален потребителски идентификатор (unique\_id), генерира се защитена парола с password\_hash(), и данните се записват в базата данни. Ако записът е успешен, се стартира потребителска сесия, като в нея се съхранява unique\_id на новия потребител. В случай на грешка при качване на файла или при вмъкване на данните в базата, скриптът връща съобщение за грешка.

Този скрипт предоставя основна, но сигурна регистрационна система, която включва валидация на входните данни, проверка за съществуващ имейл, защита срещу SQL инжекции, обработка на изображения и криптиране на пароли.



Този PHP скрипт е част от входния процес на потребителите в приложението **Real Time Chat App**. Той започва с инициализация на сесията чрез командата session\_start() и създаване на връзка с базата данни, като се включва конфигурационен файл с данни за достъп до базата. След установяване на връзката, скриптът проверява дали връзката е успешна, като ако има проблеми с нея, се извежда съобщение за грешка.

След това се проверява дали е подадена заявка чрез метод POST, което означава, че потребителят се опитва да влезе в системата. Входните данни, като имейл и парола, се извеждат от формата на вход и преминават през защита от SQL инжекции, като се използва функцията mysqli\_real\_escape\_string(). Ако не са подадени необходимите данни или е извършена невалидна заявка, се показват съобщения за грешка.

Когато данните са попълнени, скриптът изпълнява SQL заявка за търсене на потребителя в базата данни по имейл. Ако в базата данни се открие съвпадение, се извлича хешираната парола, свързана с този имейл. След това се използва функцията password\_verify() за да се провери дали въведената парола съвпада с хешираната в базата данни. Ако паролата е вярна, статусът на потребителя се актуализира на "online" в базата данни.

Ако паролата е валидна и статусът е актуализиран, се записва уникалният идентификатор на потребителя в сесията и се показва съобщение за успешен вход. Ако има някаква грешка при актуализиране на статуса или сесията, потребителят получава съобщение за неуспешен вход.

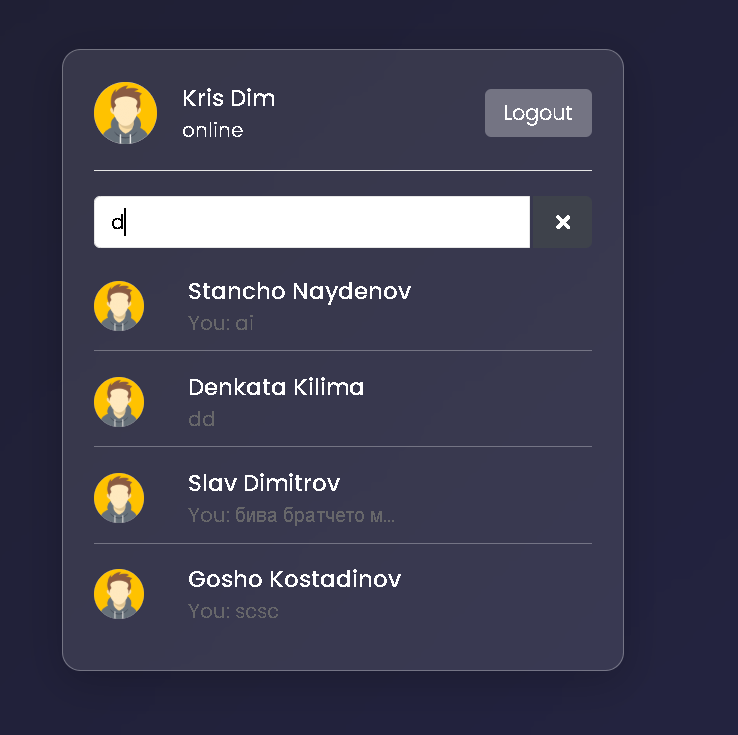
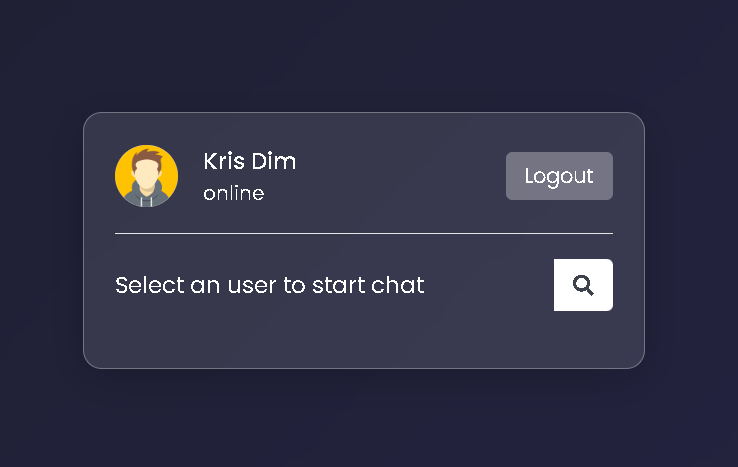
Ако имейлът не съществува в базата данни или паролата не е правилна, скриптът връща съобщения за грешка, като показва, че потребителят с такъв имейл не е открит или че имейлът и паролата не съвпадат.

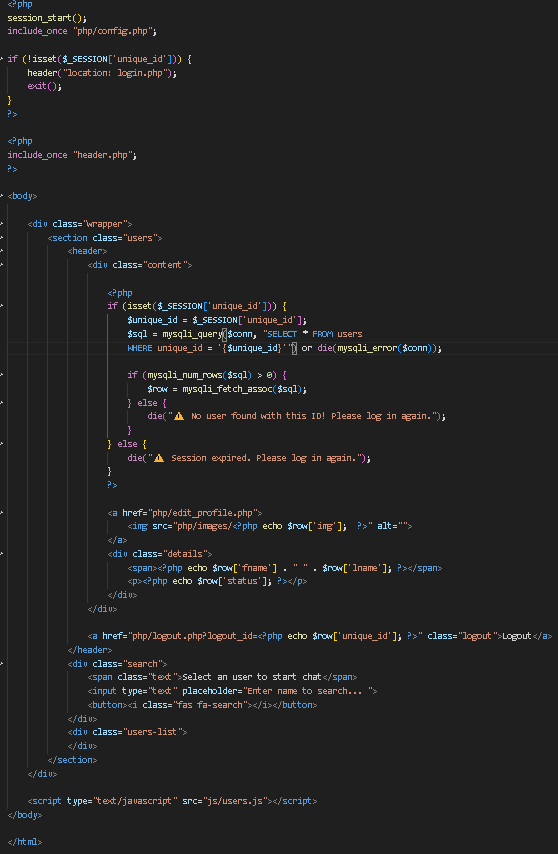
Общо взето, този скрипт осигурява основна функционалност за вход в приложението, като използва добри практики за защита на данните и предотвратяване на SQL инжекции. За да се подобри сигурността, може да се добавят допълнителни проверки, като например за защита срещу атаки чрез брут форс и управление на сесиите (например с ограничаване на сесионното време или добавяне на CAPTCHA при многократни неуспешни опити за вход).

## **Потребителски интерфейс**

Дизайнът на потребителския интерфейс играе ключова роля в успеха на всяко приложение. Системата разполага със специализиран модул за търсене на потребители.

Представлява страница за чат, която зарежда списък с потребители и показва информация за текущо влезлия потребител.





Този PHP скрипт е част от уеб приложение за чат и има за цел да осигури защитен достъп до страницата със списък на потребителите. Той започва с проверка дали има активна сесия, като използва session\_start(), за да зареди сесийните данни. Ако потребителят не е влязъл в профила си, скриптът го пренасочва към страницата за вход (login.php), като по този начин предотвратява неоторизиран достъп.

След като се увери, че потребителят е влязъл, скриптът извлича информация за него от базата данни, като използва уникалния му идентификатор (unique\_id), съхранен в сесията. Тази информация включва първото и последното име, статуса (например „Онлайн“ или „Офлайн“) и профилната снимка. Ако не бъде намерен потребител с дадения идентификатор, се показва съобщение за грешка и се изисква повторно влизане.

Интерфейсът на страницата съдържа различни функционалности за улесняване на комуникацията. Профилната снимка на потребителя служи като линк към страницата за редактиране на профила (edit\_profile.php), което му позволява да актуализира своята информация. Освен това, има бутон за изход (logout.php), който прекратява сесията и изключва потребителя от системата.

Допълнителна функционалност включва поле за търсене, което позволява на потребителите да намират други участници в чата, като въведат тяхното име. Под това поле има контейнер (.users-list), който първоначално е празен, но се запълва динамично чрез JavaScript файл (users.js). Този скрипт обработва заявките за търсене и показва списък с наличните потребители.

Скриптът осигурява сигурен и удобен начин за взаимодействие между потребителите, като ограничава достъпа само до автентифицирани потребители и динамично зарежда информация за тях. Това допринася за изграждането на ефективна и сигурна чат система.



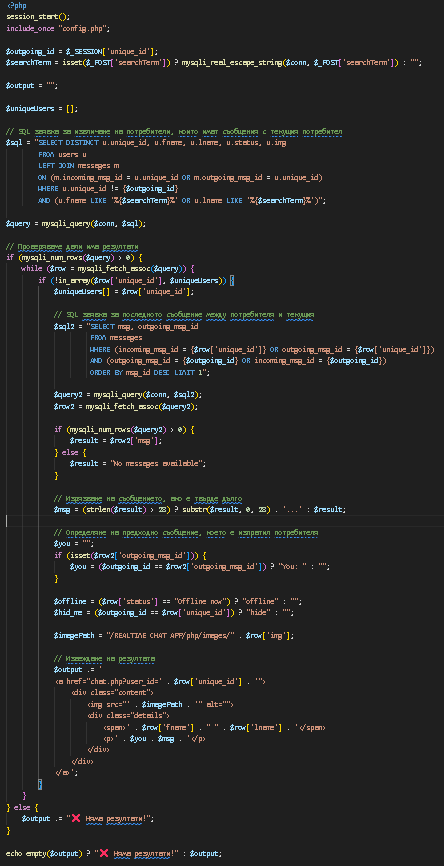
Този JavaScript код има две основни функции: да позволява търсене на потребители в реално време и да осигурява автоматично обновяване на списъка с активни потребители.

Когато потребителят кликне върху иконата за търсене, полето за търсене се показва или скрива. Ако е активно, автоматично получава фокус, за да може потребителят веднага да започне да въвежда текст. Ако потребителят натисне бутона повторно, докато полето е активно, въведената информация се изчиства, а търсенето се прекратява.

При въвеждане на текст в полето за търсене се изпраща заявка към сървъра (search.php), който обработва въведените данни и връща списък с потребители, съответстващи на търсената дума. Тези резултати динамично заменят текущия списък с потребители, без да е необходимо презареждане на страницата. Ако полето за търсене е празно, търсенето се изключва и страницата се връща в нормалния си режим.

Освен тази функция за търсене, кодът също така автоматично обновява списъка с потребители на всеки 30 секунди чрез заявка към users.php. Това гарантира, че потребителите виждат актуална информация за онлайн статусите и наличните контакти. Ако потребителят активно използва търсенето, автоматичното обновяване временно се спира, за да не презаписва резултатите от търсенето.

Тази логика осигурява удобен и интерактивен потребителски интерфейс, като комбинира динамично търсене и автоматично обновяване, за да направи работата с чат системата по-ефективна и удобна.



Този PHP скрипт представлява част от функционалността на чат приложение, което позволява на потребителите да търсят други потребители и да виждат резултати, свързани със съобщенията между тях. Скриптът започва с инициализиране на сесия и извличане на уникалния идентификатор на текущия потребител от сесията, за да може да бъде направено сравнение с останалите потребители в базата данни.

Основната му задача е да извърши търсене по име и фамилия на потребители, които вече имат обменени съобщения с текущия потребител. Параметърът за търсене (термин за търсене на име или фамилия) се получава чрез POST заявка и след това се ескейпира, за да се предотвратят SQL инжекции. Изпълнява се SQL заявка, която извлича всички потребители, с които текущият потребител е водил разговори, като се филтрират само тези, чийто име или фамилия съдържат въведения термин за търсене.

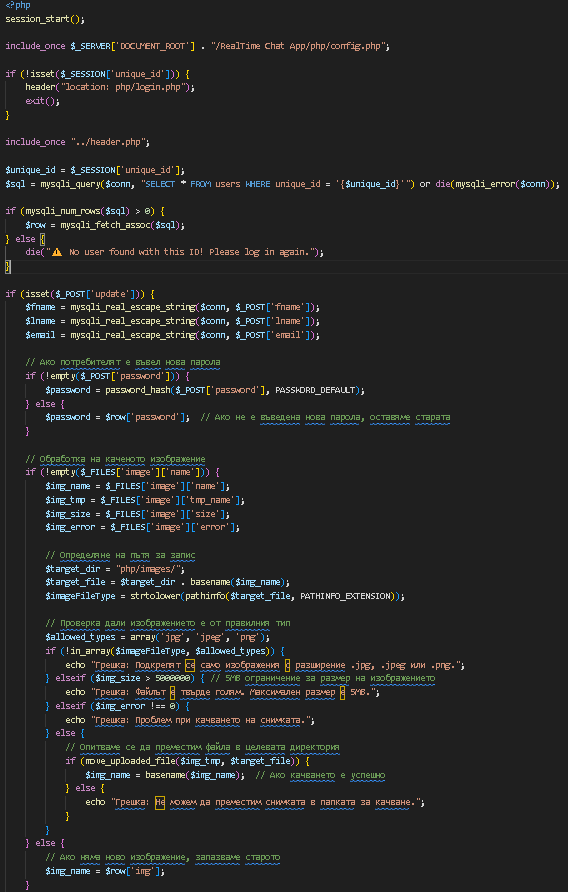
За всеки от намерените потребители се изпълнява допълнителна SQL заявка, която извлича последното съобщение между тях и текущия потребител. Това съобщение се показва в резултатите, като ако съобщението е твърде дълго, то се съкращава до 28 символа. Ако няма съобщения, се извежда текстът "No messages available". Освен последното съобщение, скриптът показва и информация за потребителя, като неговото име, фамилия, статус (например "Online" или "Offline") и профилна снимка.

Скриптът също така проверява дали текущият потребител е изпратил последното съобщение. Ако това е така, съобщението започва с текста "You: ", за да се подчертае, че то е изпратено от него. Потребителите, които вече са в чата с текущия потребител, няма да се показват повторно в резултатите от търсенето.

В края на процеса, ако са намерени резултати, те се показват като HTML елементи, включително линк към страницата за чат с всеки от тези потребители. Ако не са намерени никакви потребители, които отговарят на търсенето, скриптът връща съобщението "❌ Няма резултати!", което информира потребителя, че няма съвпадения.

Този скрипт играе важна роля в чат приложението, като осигурява функционалността за търсене на други потребители, показване на последни съобщения и предоставяне на информация за онлайн статус. Той предлага лесен начин за потребителите да намерят хора, с които са водили разговори, и да продължат комуникацията си, като едновременно с това се гарантира сигурност чрез SQL инжекции и оптимизирана работа с данни.

## **Редактирране на профил**



Този PHP код е част от функционалността на приложението **Real Time Chat App**, предназначено за редактиране на потребителски профили. Той позволява на потребителите да актуализират своите лични данни, като име, фамилия, имейл адрес, парола и профилна снимка. Процесът започва с проверка дали потребителят е влязъл в системата, като ако не е, се извършва пренасочване към страницата за вход. Това гарантира, че само автентифицирани потребители могат да правят промени в своите профили.

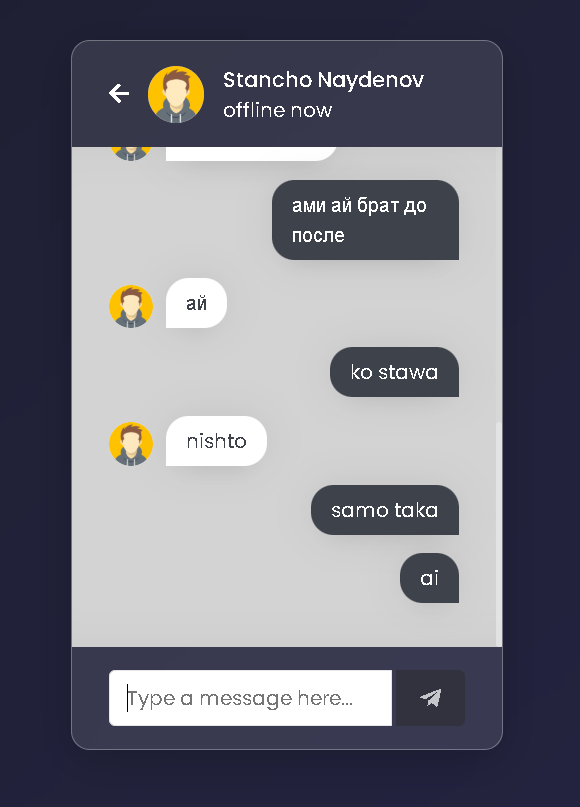
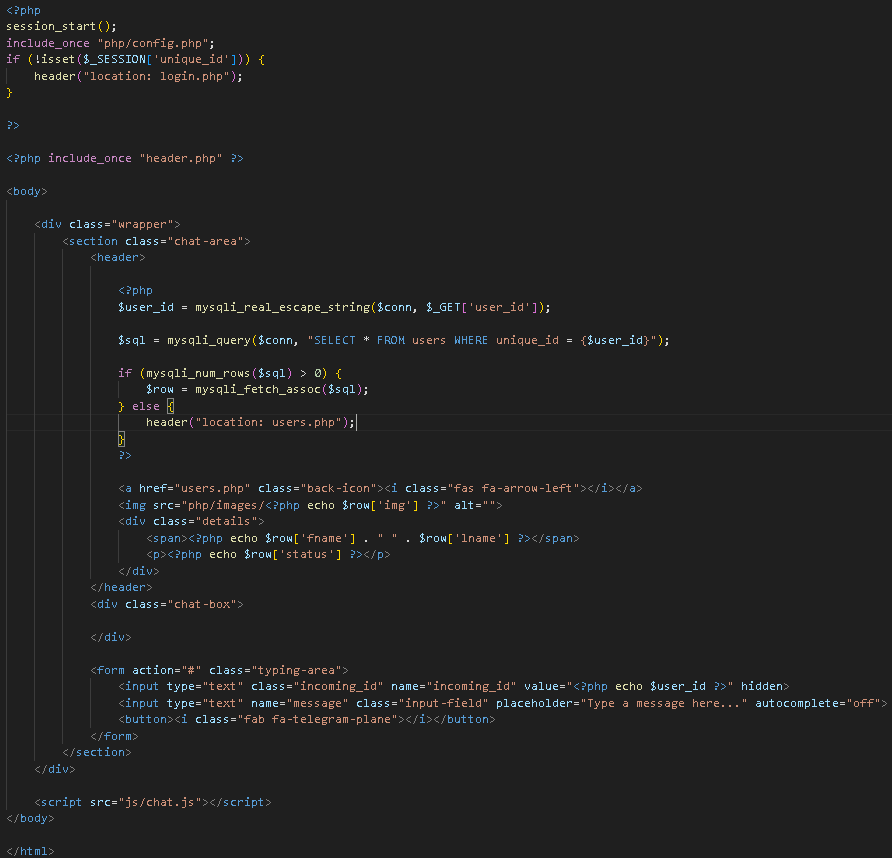
След като потребителят е потвърден, се извличат текущите му данни от базата данни и се показват в полетата на формата за редактиране. Така потребителят може да види вече попълнените си данни и да ги коригира по желание. Потребителят има възможност да въведе нова парола или да остави старата, ако не желае промяна. Ако се въведе нова парола, тя се хешира преди да бъде записана в базата данни, което осигурява сигурността на потребителските пароли.

Допълнително, ако потребителят желае да промени снимката си, може да качи ново изображение, като преди това се извършват няколко проверки за валидността на файла. Тези проверки включват проверка на типа на файла, който трябва да е от тип .jpg, .jpeg или .png, както и проверка на размера на файла, който не трябва да надвишава 5MB. Ако файлът не отговаря на тези условия, се показва съобщение за грешка.

След като потребителят подаде формата, данните се актуализират в базата данни чрез SQL заявка. Ако заявката е успешна, потребителят се пренасочва към страницата със списък на потребителите. В противен случай се показва съобщение за грешка.

Тази част от кода осигурява важна функционалност за персонализиране на профилите на потребителите, като позволява актуализиране на информацията и качване на нова снимка. Въпреки това, кодът може да бъде усъвършенстван чрез използване на подготовени SQL изявления (prepared statements), за да се избегне рискът от SQL инжекции, както и с добавяне на допълнителна валидация на въведените данни за още по-добра сигурност.

## **Chat страница**



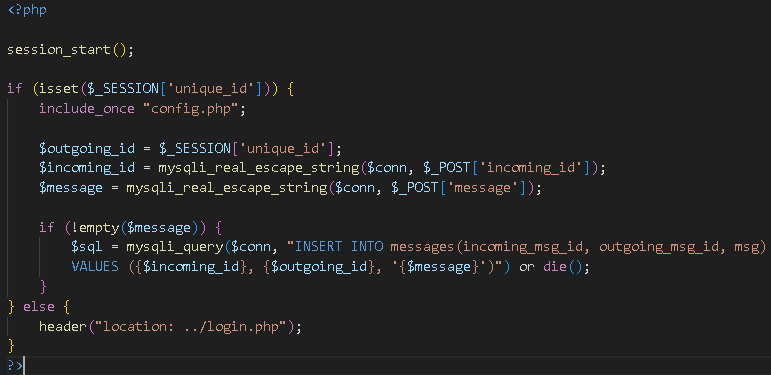
Този PHP скрипт е част от чат приложение, което осигурява възможност на потребителите да водят лични разговори помежду си. Скриптът започва с проверка дали потребителят е влязъл в системата чрез сесия. Ако не е, потребителят бива пренасочен към страницата за вход. След това извлича информация за избрания потребител (с когото ще се води чат) чрез параметъра user\_id, подаден в URL адреса. Чрез SQL заявка се получава информация за събеседника, включително неговото име, фамилия, статус и профилна снимка. Ако потребителят с този идентификатор не съществува в базата данни, той бива пренасочен към списъка с всички потребители.

На страницата се показва информация за събеседника – неговото име, статус и снимка. Освен това се предоставя форма, в която потребителят може да въведе съобщение, което да изпрати на избрания събеседник. Идентификаторът на получателя е скрит в полето на формата, за да може съобщението да бъде изпратено правилно. Когато съобщението бъде изпратено, се показва бутон за изпращане, като използваната икона е типична за чат приложенията.

Скриптът използва функцията mysqli\_real\_escape\_string(), за да защити данните срещу SQL инжекции, което е важно за сигурността на приложението. Това гарантира, че всички въведени стойности за параметрите на заявките ще бъдат безопасно обработени, без да се допуска манипулация на SQL заявките.

Ако по някаква причина няма намерен съответен потребител с подаденото user\_id, потребителят се пренасочва обратно към списъка с всички потребители. Това осигурява правилно функциониране на приложението и предотвратява показването на празни или некоректни страници.

Целият процес е насочен към предоставяне на безопасна и лесна за използване среда за комуникация между потребителите, като се осигурява сигурност, добра навигация и лесен достъп до функционалността за чат.

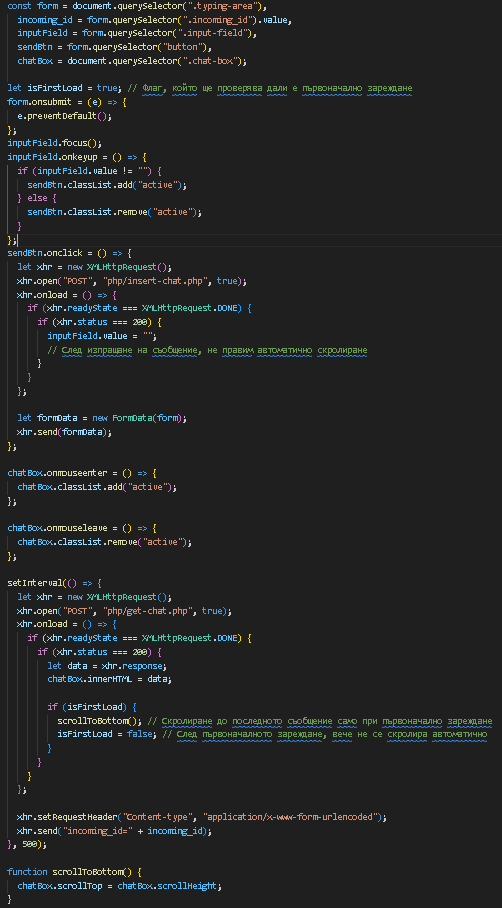


Тези два PHP скрипта представляват основната функционалност на чат система, която позволява на потребителите да обменят съобщения помежду си в реално време.

Първият скрипт е отговорен за показването на съществуващите съобщения между двама потребители. Той започва с проверка дали потребителят е влязъл в системата чрез проверка на сесията. Ако сесията е активна, скриптът извлича всички съобщения, които са изпратени между текущия потребител и избрания събеседник, като ги подрежда по дата на изпращане. Съобщенията се показват с различен стил в зависимост от това дали са изпратени от текущия потребител или от събеседника. Ако няма съобщения, се показва съобщение, че няма налични съобщения и се насърчава потребителят да изпрати съобщение.

Вторият скрипт позволява изпращането на ново съобщение. Той отново проверява дали потребителят е влязъл в системата, и ако това е така, взема данни за получателя и съобщението чрез POST заявката. След това съобщението се вмъква в базата данни, така че да бъде достъпно за последващи извличания и показване. Ако сесията не е активна, потребителят бива пренасочен към страницата за вход.

Тези два скрипта работят заедно, за да осигурят основната функционалност на чат системата, като обработват както показването на съществуващите съобщения, така и изпращането на нови. Скриптовете осигуряват защита срещу SQL инжекции чрез използването на функцията mysqli\_real\_escape\_string() за обработка на входни данни.



Този JavaScript код управлява основната функционалност на чат система в уеб приложение, като осигурява изпращане на съобщения, обновяване на чат прозореца и автоматично скролиране до последното съобщение. Кодът е проектиран да работи с асинхронни HTTP заявки, което прави комуникацията бърза и без прекъсвания, без да е необходимо презареждане на страницата.

Първата важна част от функционалността е предотвратяването на изпращането на формата чрез стандартната препращаща операция, използвайки e.preventDefault() в обработчика за събитие на изпращане на формата. Това позволява данните да се изпратят асинхронно, без да се презарежда страницата. След това, ако потребителят започне да пише съобщение в полето за въвеждане, бутонът за изпращане става активен, а ако полето е празно, бутонът се деактивира. Това подобрява потребителския интерфейс, като предотвратява изпращането на празни съобщения.

Когато потребителят натисне бутона за изпращане, се създава нова асинхронна HTTP заявка чрез XMLHttpRequest. Тази заявка изпраща съдържанието на формата към сървъра, като съобщението се записва в базата данни. След успешното изпращане на съобщението, полето за въвеждане се изчиства, но не се извършва автоматично скролиране към последното съобщение в чата.

Чат прозорецът се обновява на всеки 500 милисекунди чрез нова асинхронна заявка, която извлича последните съобщения от сървъра и ги показва в чата. Ако това е първоначално зареждане на чата, се извършва автоматично скролиране до последното съобщение. Това е удобно, защото позволява на потребителя да започне да вижда съобщенията от най-новото без да се налага да търси последното съобщение ръчно.

Кодът също така добавя визуални ефекти за подобряване на потребителското изживяване. Когато курсорът на мишката е върху чат прозореца, прозорецът получава клас "active", което може да води до активиране на различни стилови промени, като показване на допълнителна информация или визуални ефекти. Когато курсорът напусне прозореца, този клас се премахва.

По този начин, този JavaScript код осигурява гладко и ефективно изживяване при комуникацията в чат приложението, като позволява бързо обновяване на съобщенията в реално време, лесно изпращане на нови съобщения и автоматично скролиране към последното съобщение, което прави интерфейса по-интуитивен и удобен за потребителите.

## **CSS**

CSS стила създава визуално атрактивен и модерен дизайн за чат приложение, като осигурява отлична функционалност и приятна за потребителя визуална атмосфера. Основните елементи на дизайна са базирани на съвременни тенденции в уеб дизайна, като използване на прозрачни фонове и размазани ефекти, които създават усещане за дълбочина и фокус върху самото съдържание. Чат прозорците и всички взаимодействия с тях са изчистени и ясни, което прави приложението лесно за употреба, без да натоварва потребителя с излишни елементи.

Когато става въпрос за съобщенията, има ясна визуална диференциация между изпратените и получените съобщения. Съобщенията на потребителя се показват със специфичен фон и цветове, докато получените съобщения имат различен стил. Това позволява лесно разграничаване на съобщенията и подобрява потребителското изживяване, като прави чат интерфейса интуитивен. Фокусирането на въведените текстови полета е подчертано чрез нежни преходи и промени във фона, което добавя допълнителна динамика към приложението.

Дизайнът е максимално опростен, с фокус върху минимализма, като всичко ненужно е премахнато, за да се осигури по-голяма яснота и ефективност. Всеки елемент, като полета за въвеждане, бутони и линкове, е с лесен за възприемане размер и разстояние, което прави навигацията много по-приятна. Елементите са ясно маркирани и подредени, а всички функционални бутони и икони са лесно достъпни, като се намират на правилните места за потребителя.

Особено внимание е отделено на адаптивността на дизайна за мобилни устройства. Чрез използването на медийни заявки (media queries) е осигурена оптимизация за малки екрани, като всички елементи се преработват така, че да изглеждат и функционират перфектно на мобилни устройства. Размерите на шрифтовете и полетата се коригират, а разстоянията между елементите се намаляват за да се оптимизира видимостта и функционалността на малки екрани.

Един от акцентите на дизайна е използването на минимални и стилни скролбарове, които не заемат излишно място, но същевременно са лесни за употреба. Стилът на самите скролбарове се настройва, така че да съответства на цялостния дизайн на приложението и да изглеждат стилно и естетично. За да се постигне гладкото и бързо превъртане на съобщенията, е предвиден плавен ефект за скролиране.

Също така, за да се подобри визуализацията, се използват различни фонови изображения, които се настройват, за да съответстват на темата на приложението. Това добавя към цялостната визуална идентичност на чат приложението и поддържа професионален и модернизиран вид.

### **Основен фон и стил на тялото(body)**

Този код задава основния фон на страницата, като комбинира изображение с линейна градиентна обработка. Използва се backdrop-filter с blur(10px), което прави задния фон размазан и придава по-съвременен и модерен вид на интерфейса.

body {

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

background: url('../images/background.jpg') no-repeat center center/cover, linear-gradient(135deg, #1e1e2f, #252545);

backdrop-filter: blur(10px);

}

### **Прозрачен и модерен дизайн за формата**

Формата (например за вход или регистрация) има прозрачен фон с размазано задно изображение, което създава ефекта за дълбочина. backdrop-filter: blur(20px) допринася за размазания ефект, а box-shadow добавя меко сянка около елемента.

.form {

background: rgba(255, 255, 255, 0.1);

backdrop-filter: blur(20px);

border-radius: 15px;

padding: 25px;

width: 400px;

box-shadow: 0 8px 32px rgba(0, 0, 0, 0.2);

border: 1px solid rgba(255, 255, 255, 0.3);

color: white;

}

### **Текстови полета с ефект при фокусиране**

Това задава стила за текстовите полета във формата. При фокусиране на полето (когато потребителят започне да пише), фона на полето леко се променя, за да се подчертае, че полето е активно.

.field input {

width: 100%;

padding: 12px;

border-radius: 8px;

border: none;

background: rgba(255, 255, 255, 0.2);

color: white;

outline: none;

transition: all 0.3s ease-in-out;

}

.field input:focus {

background: rgba(255, 255, 255, 0.3);

}

### **Съобщения в чат (изпратени и получени)**

В този фрагмент се определят стиловете за изпратени (outgoing) и получени (incoming) съобщения в чата. Изпратените съобщения имат тъмен фон с бял текст, а получените съобщения имат светъл фон с тъмно сив текст.

.chat-box .outgoing .details p {

background: #3e424b;

color: #fff;

border-radius: 18px 18px 0 18px;

}

.chat-box .incoming .details p {

background: #fff;

color: #3e424b;

border-radius: 18px 18px 18px 0;

}

### **Типично поле за въвеждане с бутон за изпращане**

Това е пример за полето за въвеждане и бутона за изпращане на съобщение в чата. Когато потребителят започне да пише в полето за въвеждане, бутонът за изпращане става активен и променя своята непрозрачност (opacity).

.typing-area input {

height: 45px;

width: calc(100% - 58px);

font-size: 16px;

padding: 0 13px;

border: 1px solid #e6e6e6;

outline: none;

border-radius: 5px 0 0 5px;

}

.typing-area button {

color: #ffffff;

width: 55px;

border: none;

outline: none;

background: #2d2f35;

font-size: 19px;

cursor: pointer;

opacity: 0.7;

pointer-events: none;

border-radius: 0 5px 5px 0;

}

.typing-area button.active {

opacity: 1;

pointer-events: auto;

}

### **Стил за списъка с потребители**

Съобщенията за потребителите, показани в списък, са стилизирани да имат кръгли аватари и добре подреден текст. Всеки елемент от списъка е линк, който има плавни преходи при взаимодействие с мишката.

.users-list a {

padding-bottom: 10px;

margin-bottom: 15px;

padding-right: 15px;

border-bottom: 1px solid rgba(255, 255, 255, 0.3);

display: flex;

align-items: center;

text-decoration: none;

transition: 0.3s;

}

.users-list a img {

height: 40px;

width: 40px;

border-radius: 50%;

margin-right: 15px;

}

### **Адаптивен дизайн за мобилни устройства**

Използват се медийни заявки за адаптиране на изгледа за мобилни устройства. Например, вътрешните отстояния и размера на шрифтовете се регулират, за да се постигне удобен за използване интерфейс на по-малки екрани.

@media screen and (max-width: 450px) {

.form,

.users {

padding: 20px;

}

.chat-box {

min-height: 400px;

padding: 10px 15px 15px 20px;

}

.chat-box .chat p {

font-size: 15px;

}

}

# **Заключение**

Приложението **Real Time Chat App** е пример за ефективно изграждане на съвременно уеб приложение, което осигурява не само функционалност, но и привлекателен и лесен за използване интерфейс. Използваните CSS техники, като backdrop-filter и динамични преходи, придават на приложението модерен и естетически издържан вид, който осигурява приятен визуален опит на потребителите.

Чат системата в **Real Time Chat App** предлага лесна и удобна навигация, като осигурява ясно разграничаване между входящите и изходящите съобщения, което улеснява потребителя в разпознаването на контекста на разговора. Формите за вход и бутоните са интуитивни и адаптирани към потребителските нужди, като се акцентира на плавни преходи и анимации, които допринасят за приятно потребителско изживяване.

Приложението е напълно адаптивно и мобилно оптимизирано, което го прави достъпно и функционално на всяко устройство, било то мобилно или десктоп. Мобилната оптимизация е ключова за **Real Time Chat App**, като гарантира, че всички функции и визуални елементи работят безпроблемно независимо от размера на екрана.

В заключение, **Real Time Chat App** успешно съчетава съвременен дизайн с функционални възможности, осигурявайки на потребителите стабилно и приятно чат изживяване. Приложението предлага интуитивен и лесен за употреба интерфейс, което го прави не само функционално, но и визуално привлекателно решение за всяко уеб приложение в съвременния дигитален свят.