

Курсов проект по дисциплината "Web програмиране"



Технически Университет Варна
Специалност: Софтуерни и Интернет технологии
курс: 3-ти

Разработващ екип:

№	Име, презиме и фамилия	Фак. №	Роля
1.	Кристиян Красимиров Писев	22621663	Ръководител на проекта и тестер. Разработване на Help модул и ръководство за потребителя. Софтуерен Разработчик

Срок за представяне: до края на семестъра 10.01.2025

Водещ преподавател: /доц. В. Божикова/

2. Задание за разработка

11. Система за предлагане и намиране на работа

В този проект ще създадете уеб приложение, което свързва работодатели и потенциалните кандидати. По същество това би било социална мрежа, която позволява на хората да общуват помежду си и по този начин да намират подходящи възможности.

Можете да черпите знания от големи социални медийни платформи като LinkedIn # Facebook. Facebook е базиран на РНР, така че можете лесно да имитирате някой от неговите функции.

Платформата ще позволи на наемателите да публикуват свободно място за работа заедно с описанието и други подробности. Системата ще изисква следните функции, за да работи оптимално:

- Прозорец за регистрация, в който човек може да се присъедини към мрежата
- Възможност за отговаряне на публикации, за да могат хората да споделят мнението си чрез коментари или други методи (като „Харесвания“)

Опция за публикуване на изображения

Администраторът ще бъде отговорен да позволи на човек да създаде акаунт в платформата.

- Система за чат, така че един потребител да може да говори с друг.

И др.

2.1. Функционални изисквания:

Системата е проектирана да осигури интуитивна и достъпна платформа за управление на обяви за работа, която обслужва два вида потребители: автори на обяви (работодатели) и читатели (потребители, които търсят работа). Основните функционалности на системата включват:

1. Управление на потребителите:

- Потребителите могат да създават акаунти чрез форма за регистрация, предоставяйки необходима информация като потребителско име, парола и имейл адрес.
- Системата предлага унифицирана форма за логване, която включва въвеждане на потребителско име (или имейл) и парола.
- След логване потребителите получават достъп до персонализиран профил, където могат да управляват своите данни.

2. Публикуване на обяви за работа:

- Работодателите могат да създават нови обяви за работа чрез форма, която включва информация като заглавие, описание, телефон за контакт, снимка.
- Всички публикувани обяви се показват в публична секция на сайта, достъпна за всички ОТОРИЗИРАНИ потребители.

3. Взаимодействие между потребителите:

Системата позволява на потребителите да разглеждат и взаимодействат с публикуваните обяви:

- Читателите могат да отбелязват харесвания („Likes“) на обяви, за да покажат интерес към тях. Могат и да коментират в полето за коментари.
- Възможност за комуникация чрез вграден чат, където потребителите могат да се свързват с работодатели или други потребители.

4. Административни функции:

- Администраторите имат достъп до специализиран административен интерфейс, където могат да наблюдават активността в системата, включително публикуваните обяви и регистрираните потребители.
- Администраторите могат да трият неподходящи обяви или да трият потребители, или техните коментари.

5. База данни:

- Системата съхранява информация за:
 - Потребители (профили с роля: администратор, работодател/читател).
 - Обяви за работа (заглавие, описание, снимка, телефон, дата на публикуване и автор).
 - Комуникация между потребители (харесвания, коментари, съобщения в чата).

2.2. Не-функционални изисквания:

Сигурност и автентикация:

- Всички пароли се съхраняват в базата данни в криптиран вид, за да се гарантира защита на личните данни.
- Достъпът до функциите на платформата се ограничава на база роля на потребителя (администратор, регистриран потребител, нерегистриран потребител).

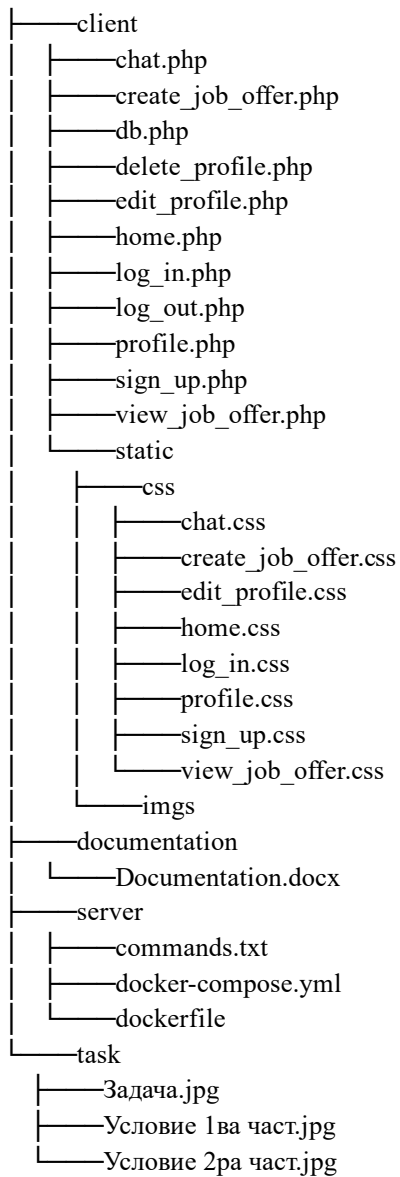
☐ Унифициран интерфейс за управление:

- Системата предоставя консистентен интерфейс за извършване на стандартни операции като създаване, преглед, редактиране и изтриване на данни.
- Всеки потребител има достъп до един и същ вид форма за управление.

☐ Надеждност:

- Системата използва релационна база данни (MySQL) за осигуряване на надеждност при съхранение и извличане на данни.

3. Реализация на проекта:



Най-важната част от проекта са базите от данни. Те се извиква във всяка една страница за да проверяват дали потребителя е регистриран или не, дали има достъп или не, какви коментари и харесвания е дал, какви обяви е създал и т.н. За избягване на постоянно свързване в с базата от данни, тя се извиква от файла „db.php“

```
<?php
session_start();
require 'db.php';
```

който представлява:

```
<?php
//echo "<h1>Testing ...</h1>";

$servername = "db";
$username = "kristiyanp";
$password = "phpproject";
$dbname = "phpdb";

$conn = new mysqli($servername, $username, $password, $dbname);

if($conn -> connect_error){
    die("Connection failed: " . $conn->connect_error);
}

//echo "Database connected sucessfully";

?>
```

Docker Compose файлът описва система, включваща PHP сървър с Apache, свързан с MySQL база данни и phpMyAdmin за управление, като всички компоненти се стартират в отделни контейнери, свързани чрез споделена мрежа с предварително зададени настройки за достъп и връзка.

```
1  services:
2      # Описание на web сървъра
3      web:
4          image: php:8.1-apache
5          container_name: php-web-server
6          volumes:
7              - ../client:/var/www/html # Връзване на директорията към сървъра
8          ports:
9              - "8080:80"
10         environment:
11             - MYSQL_HOST=db
12             - MYSQL_USER=kristiyanp
13             - MYSQL_PASSWORD=phpproject
14             - MYSQL_DATABASE=phpdb
15         depends_on: # Базата преди сървъра
16             - db
17         command: /bin/sh -c "docker-php-ext-install mysqli && apache2-foreground" # Не ми тръгваше без това
18         networks:
19             - app-network
20
21     db:
22         image: mysql:5.7
23         container_name: mysql-db
24         environment:
25             MYSQL_ROOT_PASSWORD: phpproject
26             MYSQL_DATABASE: phpdb
27             MYSQL_USER: kristiyanp
28             MYSQL_PASSWORD: phpproject
29         ports:
30             - "3306:3306"
31         networks:
32             - app-network
33
34     phpmyadmin:
35         image: phpmyadmin/phpmyadmin
36         container_name: phpmyadmin
37         environment:
38             PMA_HOST: mysql-db
39             PMA_USER: kristiyanp
40             PMA_PASSWORD: phpproject
41         ports:
42             - "8081:80"
43         depends_on:
44             - db
45         networks:
46             - app-network
47
48     networks:
49         app-network:
50             driver: bridge
```

Всяка една страница от сайта, в която достъпа е ограничен за неоторизирани потребители се проверява с една и съща проверка, която препраща тези потребители за да се включат в системата:

```
if (!isset($_SESSION['user_id'])) {  
    header("Location: log_in.php");  
    exit;  
}
```

Оторизираните лица имат достъп до следните страници (и администраторите има допълнителен достъп до администрацията):

```
<div class="nav-right">  
    <?php if ($is_logged_in): ?>  
        <a href="profile.php">Profile</a>  
        <a href="chat.php">Chat</a>  
        <a href="log_out.php">Log Out</a>  
        <a href="create_job_offer.php">Create Job Offer</a>  
        <?php if ($is_admin): ?>  
            <a href="http://localhost:8081/" target="_blank">Administration</a>  
        <?php endif; ?>  
    <?php else: ?>  
        <a href="log_in.php">Log In</a>  
        <a href="sign_up.php">Sign Up</a>  
    <?php endif; ?>  
</div>
```

Търсенето на информация за конкретен потребител се извършва във всички страници чрез SQL заявки:

```
$sql_users = "SELECT id, username FROM users WHERE id != ?";  
$stmt = $conn->prepare($sql_users);  
$stmt->bind_param("i", $user_id);  
$stmt->execute();  
$users_result = $stmt->get_result();  
$users = $users_result->fetch_all(MYSQLI_ASSOC);  
$stmt->close();
```


Началната страница трябва да листва всички налични обяви, също със SQL заявка и цикъл за извеждане на всяка една обява и нейните детайли:

```
$sql = "SELECT job_offers.id, job_offers.title, job_offers.description, job_offers.likes, job_offers.created_at
FROM job_offers
JOIN users ON job_offers.user_id = users.id";
$result = $conn->query($sql);
?>

<?php while ($job = $result->fetch_assoc()): ?>
    <div class="job-offer">
        <h4><a href="view_job_offer.php?job_id=<?php echo $job['id']; ?>"><?php echo htmlspecialchars($job['title'])
        <p><b>Posted by:</b> <?php echo htmlspecialchars($job['username']); ?></p>
        <p><b>Description:</b> <?php echo htmlspecialchars(substr($job['description'], 0, 100)) . '...'; ?></p>
        <p><b>Likes:</b> <?php echo $job['likes']; ?></p>
        <p>Posted on: <?php echo htmlspecialchars($job['created_at']); ?></p>
    </div>
<?php endwhile; ?>
```

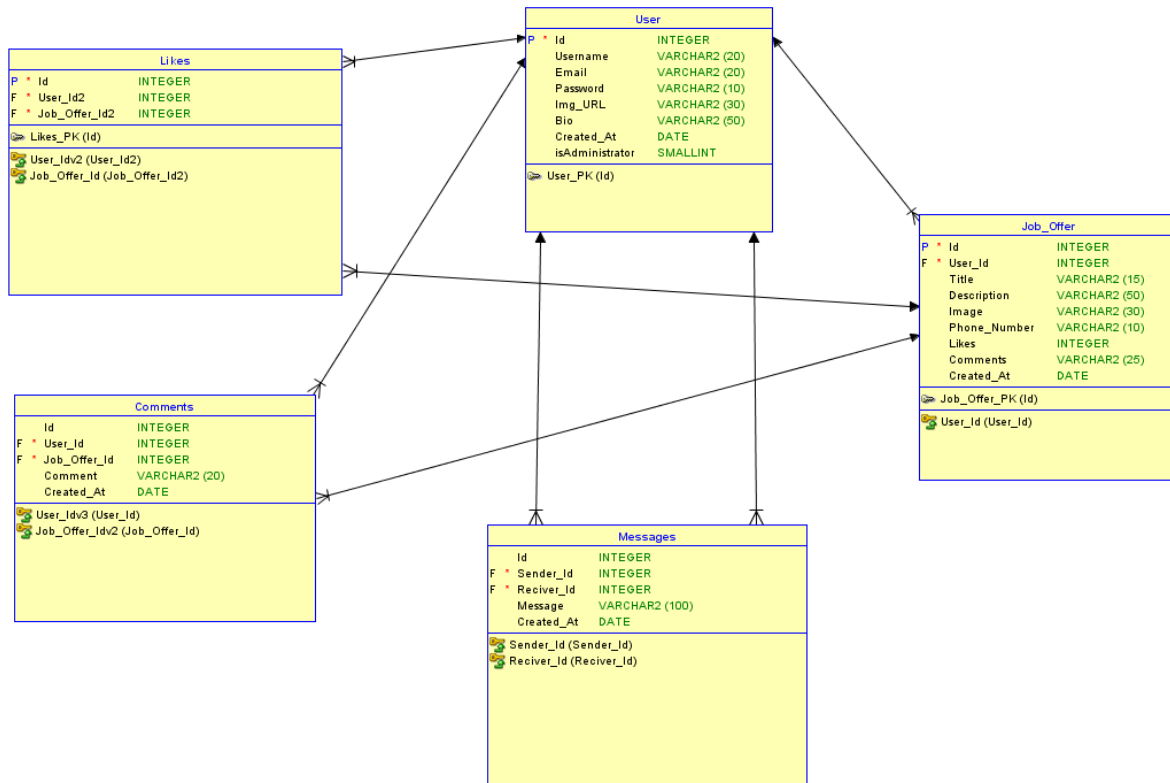
За да се избегне конфликт с информацията на потребителя при изтриване от базата от данни трябва да се изтрие всяка една релация, всичко свързано с конкретния потребител: кои обяви е хареса потребителя, неговите обяви, харесванията на неговите обяви от други потребители. По същият начин и коментарите и съобщенията

```
// Delete likes to user
$sql_delete_likes_offers = "DELETE FROM likes WHERE job_id IN (SELECT id FROM job_offers WHERE user_id = ?)";
$stmt_likes_offers = $conn->prepare($sql_delete_likes_offers);
$stmt_likes_offers->bind_param("i", $user_id);
$stmt_likes_offers->execute();
$stmt_likes_offers->close();

//Delete likes from the user
$sql_delete_likes_user = "DELETE FROM likes WHERE user_id = ?";
$stmt_likes_user = $conn->prepare($sql_delete_likes_user);
$stmt_likes_user->bind_param("i", $user_id);
$stmt_likes_user->execute();
$stmt_likes_user->close();

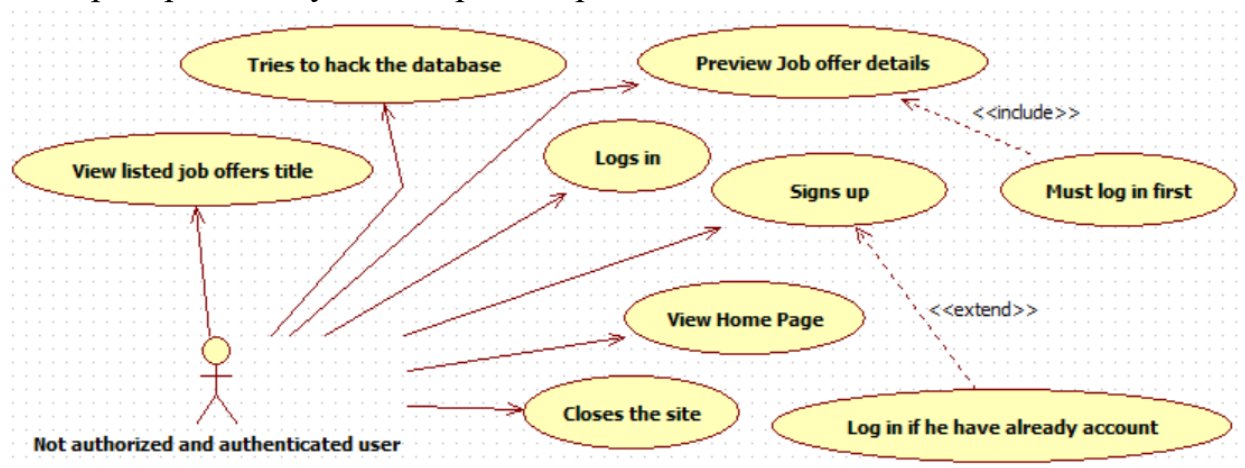
// Delete comments to user
$sql_delete_comments_offers = "DELETE FROM comments WHERE job_id IN (SELECT id FROM job_offers WHERE user_id = ?)";
$stmt_comments_offers = $conn->prepare($sql_delete_comments_offers);
$stmt_comments_offers->bind_param("i", $user_id);
$stmt_comments_offers->execute();
$stmt_comments_offers->close();
```

- Entity-Relationship Диаграма

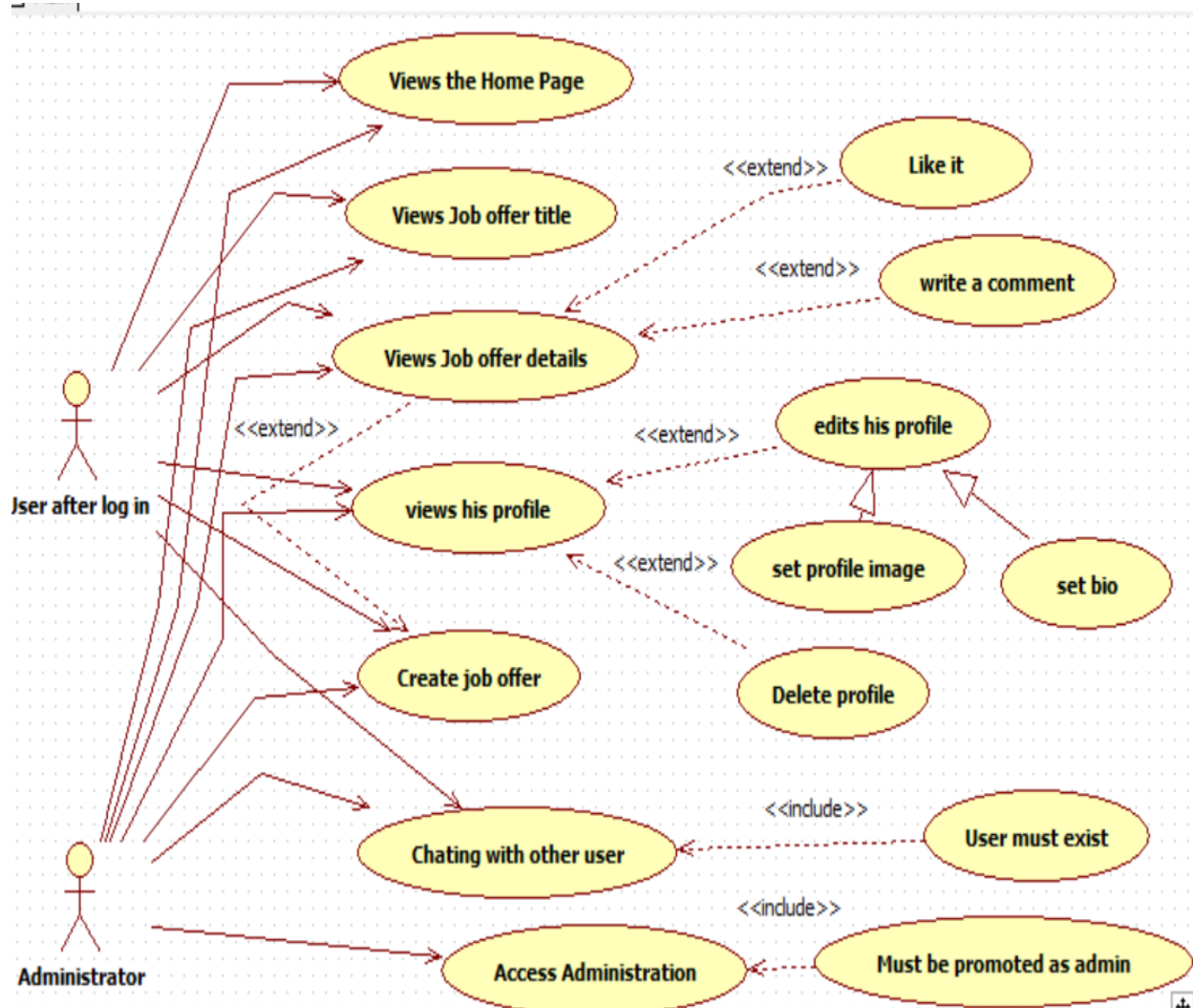


- UML Use Case Диаграми

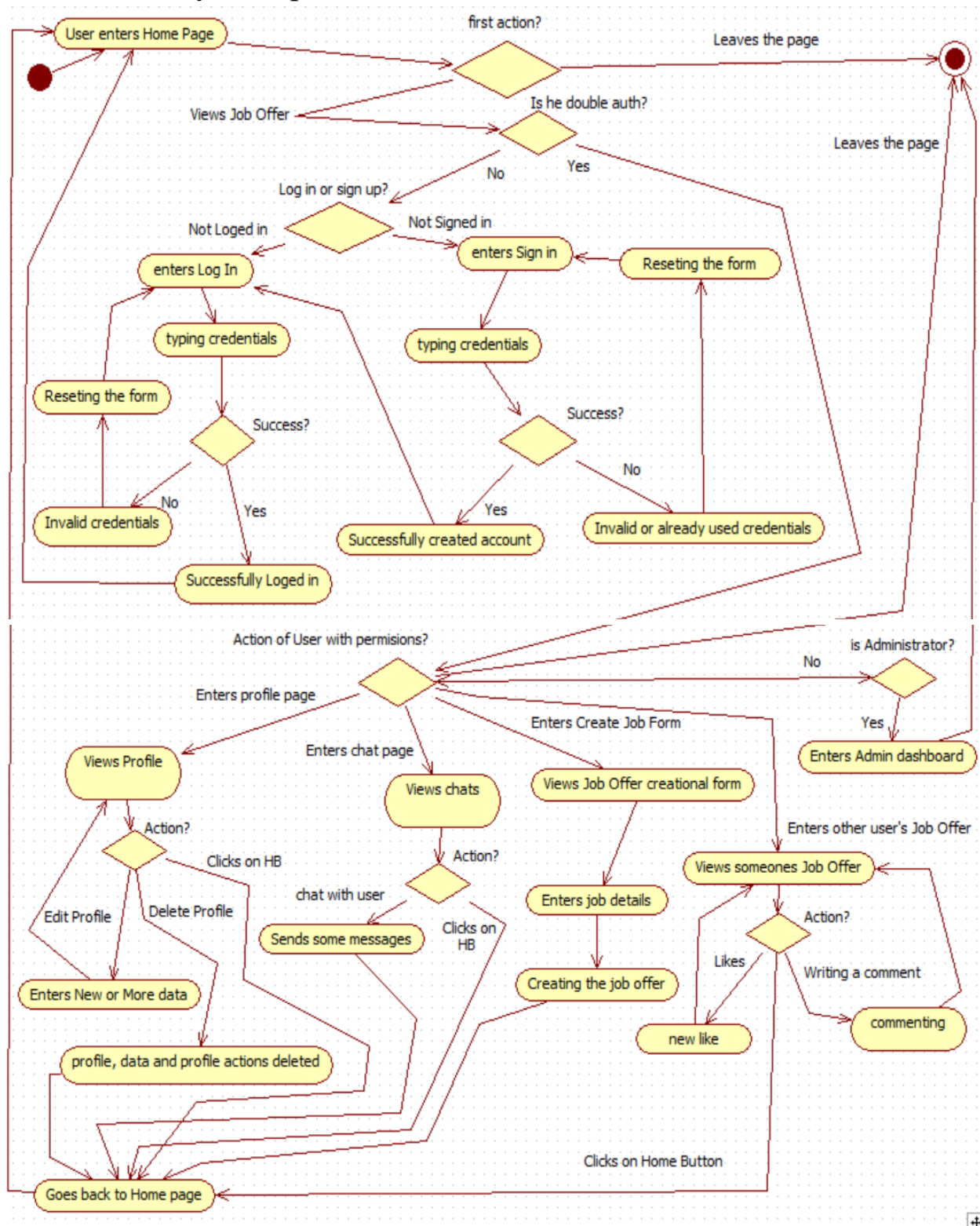
неоторизиран и неаутентикиран потребител



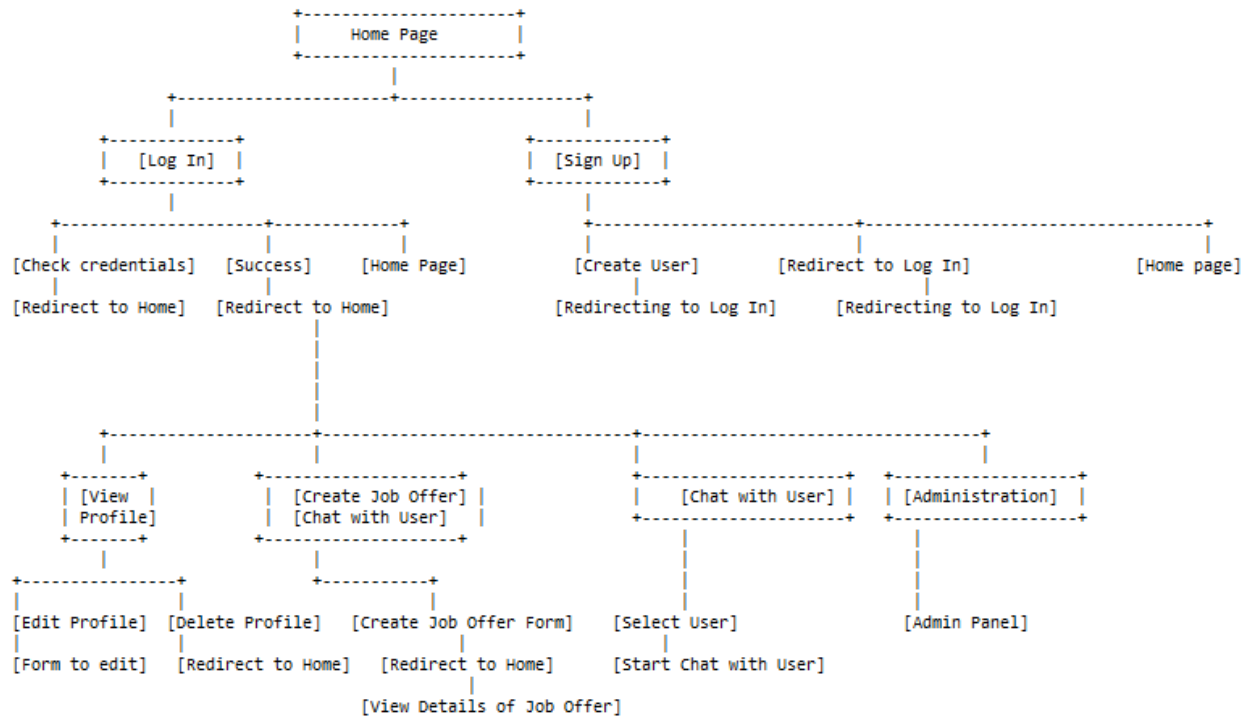
Оторизиран, аутентикиран потребител и администратор



- UML Activity Диаграма

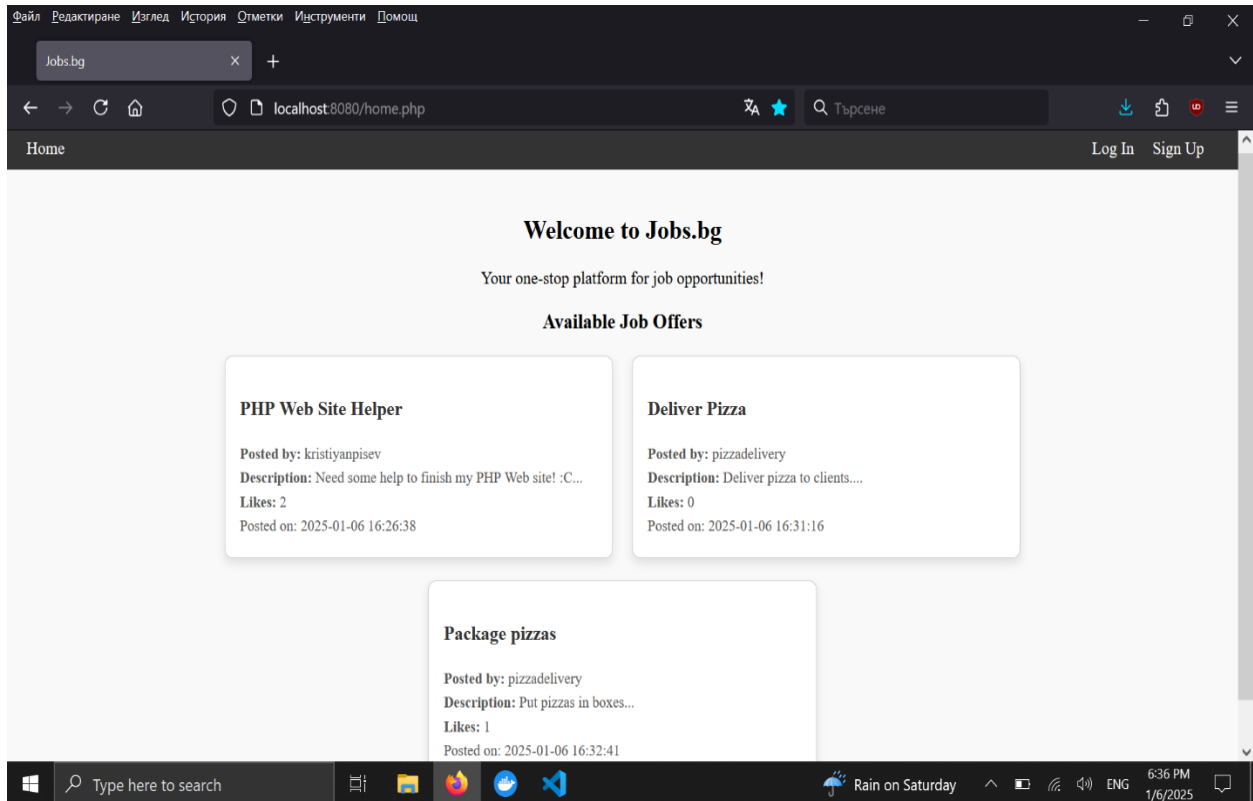


- Блок Схема



4. Кратко ръководство за потребителя. Тестване на приложението. Изводи и възможности за развитие.

Потребител се присъединява към платформата след оторизация и аутентикация (Sign up / Log in).



При избиране на една от опциите за оторизация и аутентикация (Sign up / Log in), потребителя ще бъде препратен към съответната форма от където ще се изискват „потребителско име“, „имейл“ и парола.

Файл Редактиране Изглед История Отметки Инструменти Помощ

Sign Up

localhost:8080/sign_up.php

Home

Sign Up

Username or email is already in use.

Username:

Email:

Password:

Sign Up

Type here to search

Near record

6:37 PM 1/6/2025

Имейла и потребителското име трябва да са уникални! В противен случай няма да бъде оторизиран потребителя. При успешна оторизация се изисква аутентикация!

Файл Редактиране Изглед История Отметки Инструменти Помощ

Log In

localhost:8080/log_in.php

Home

Log In

Username or Email:

Password:

Log In

Don't have an account? [Sign Up](#)

localhost:8080/sign_up.php

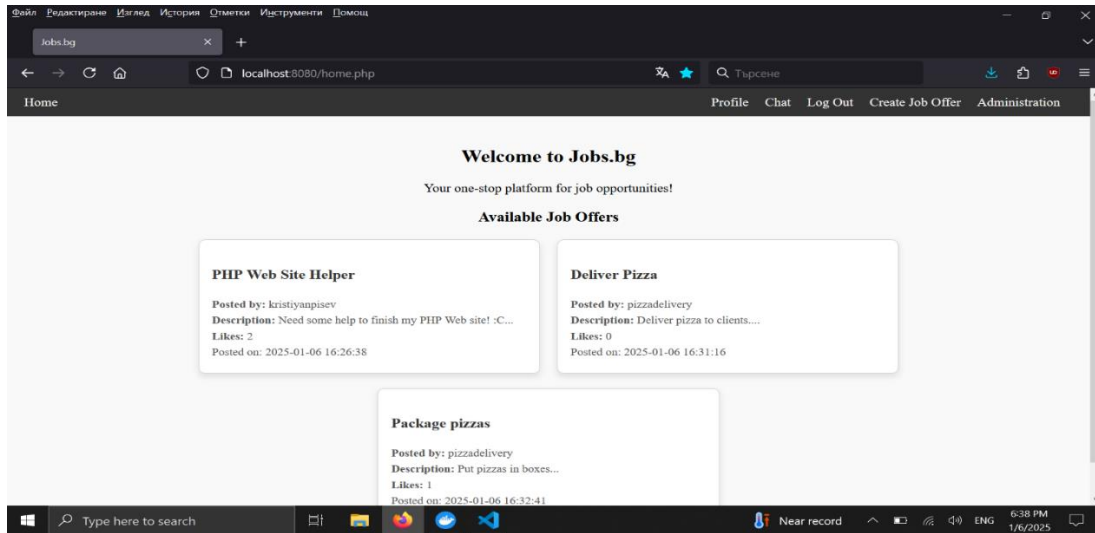
Type here to search

Rain on Saturday

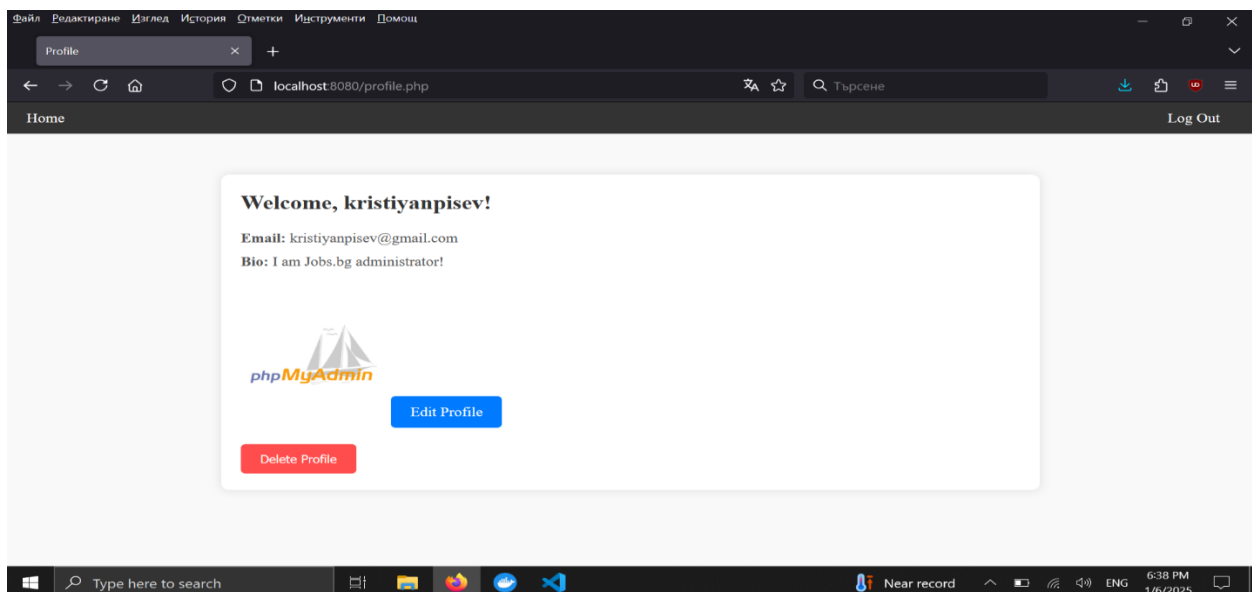
6:36 PM 1/6/2025

След успешно влизане в системата нови опции се показват и разрешават на главната страница:

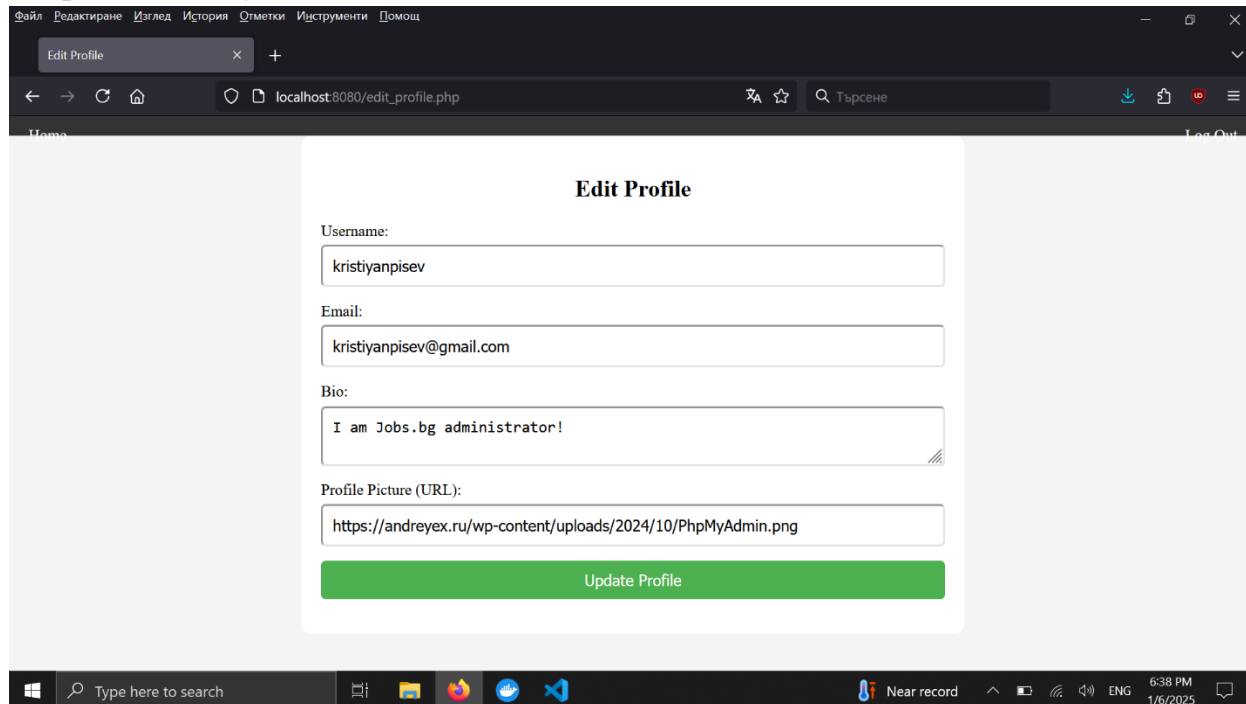
- Профил (за персонализиране на потребителя)
- Чат (за комуникация с други потребители)
- Създаване на оферта (създаване на реклама за работа)
- Администрация САМО ЗА административни лица



В страницата „Профил“ се съдържа информация за потребителя: Име, Емейл, Био и персонализирана снимка (линк). Провила може да бъде актуализирана по собствен вкус или при нужда изтриван от системата.



Опцията „Редактирай Профил“ прехвърля потребителя до следната форма за редакция на профила. Опцията „Изтрий Профил“ изтрива всичката информация за потребителя и го изхвърля от системата очаквайки нова авторизация и аутентикация.

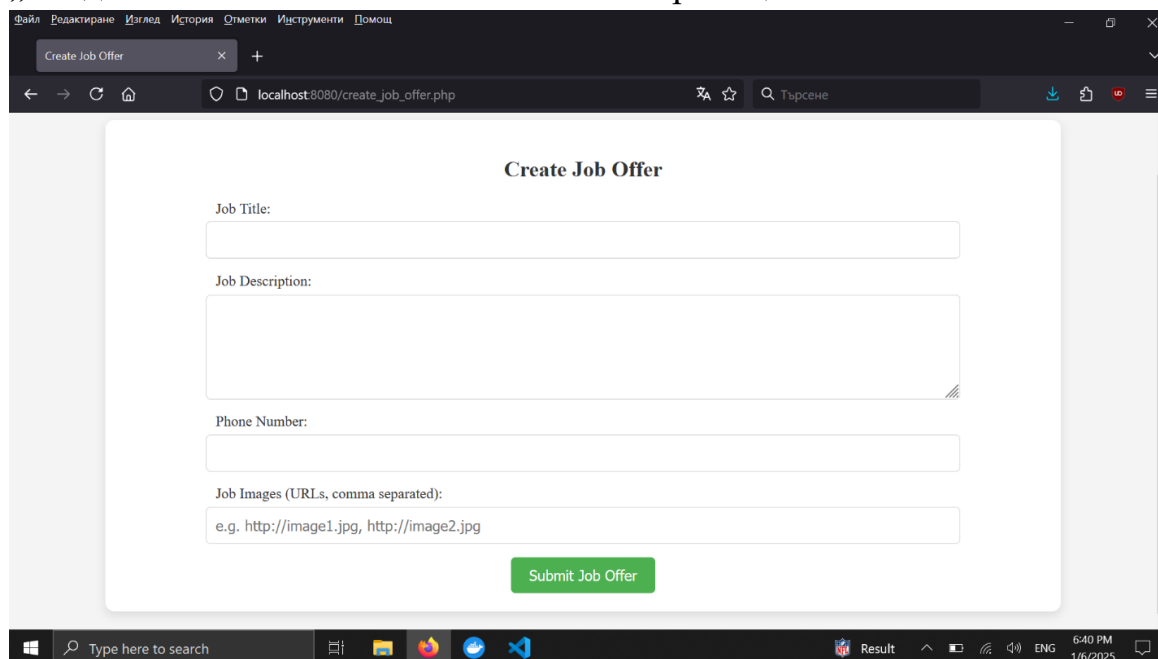


The screenshot shows a web browser window with the address bar displaying 'localhost:8080/edit_profile.php'. The page title is 'Edit Profile'. The form contains the following fields:

- Username:** A text input field containing 'kristiyanpisev'.
- Email:** A text input field containing 'kristiyanpisev@gmail.com'.
- Bio:** A text area containing 'I am Jobs.bg administrator!'.
- Profile Picture (URL):** A text input field containing 'https://andreyex.ru/wp-content/uploads/2024/10/PhpMyAdmin.png'.

At the bottom of the form is a green button labeled 'Update Profile'.

Потребителя може да публикува собствени обяви достъпвайки опцията „Създай Оява за Работа“ от началната страница.

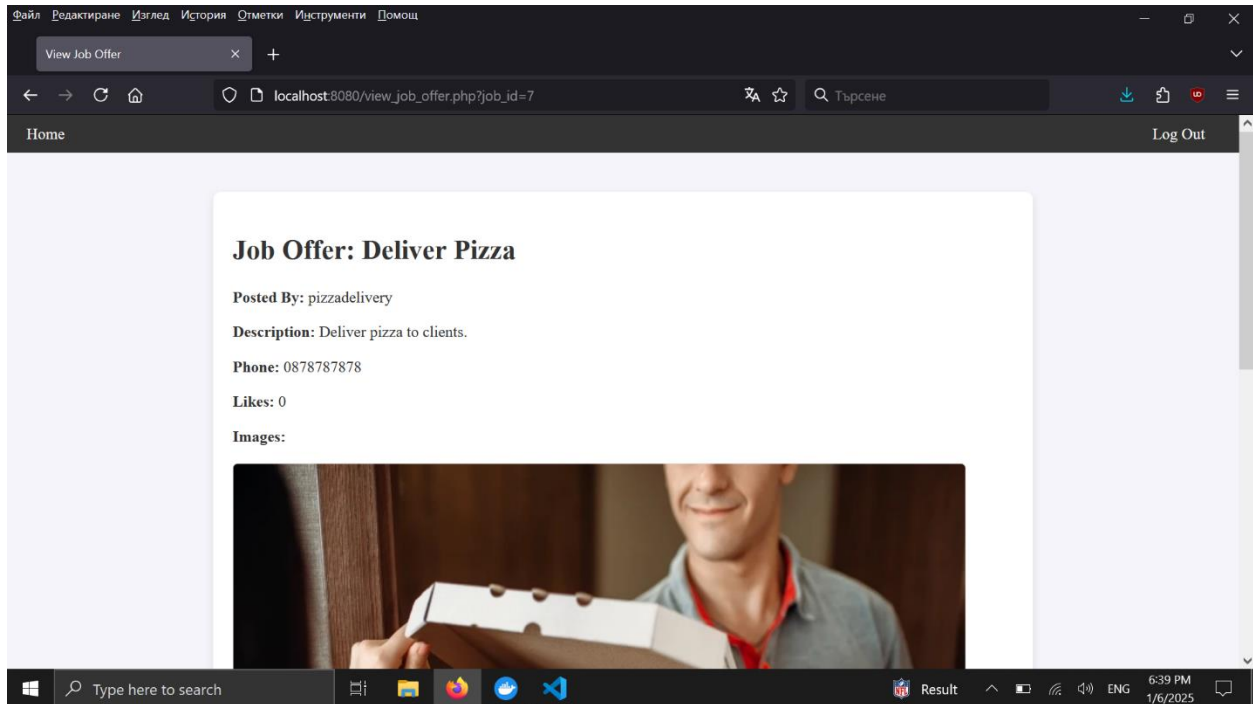


The screenshot shows a web browser window with the address bar displaying 'localhost:8080/create_job_offer.php'. The page title is 'Create Job Offer'. The form contains the following fields:

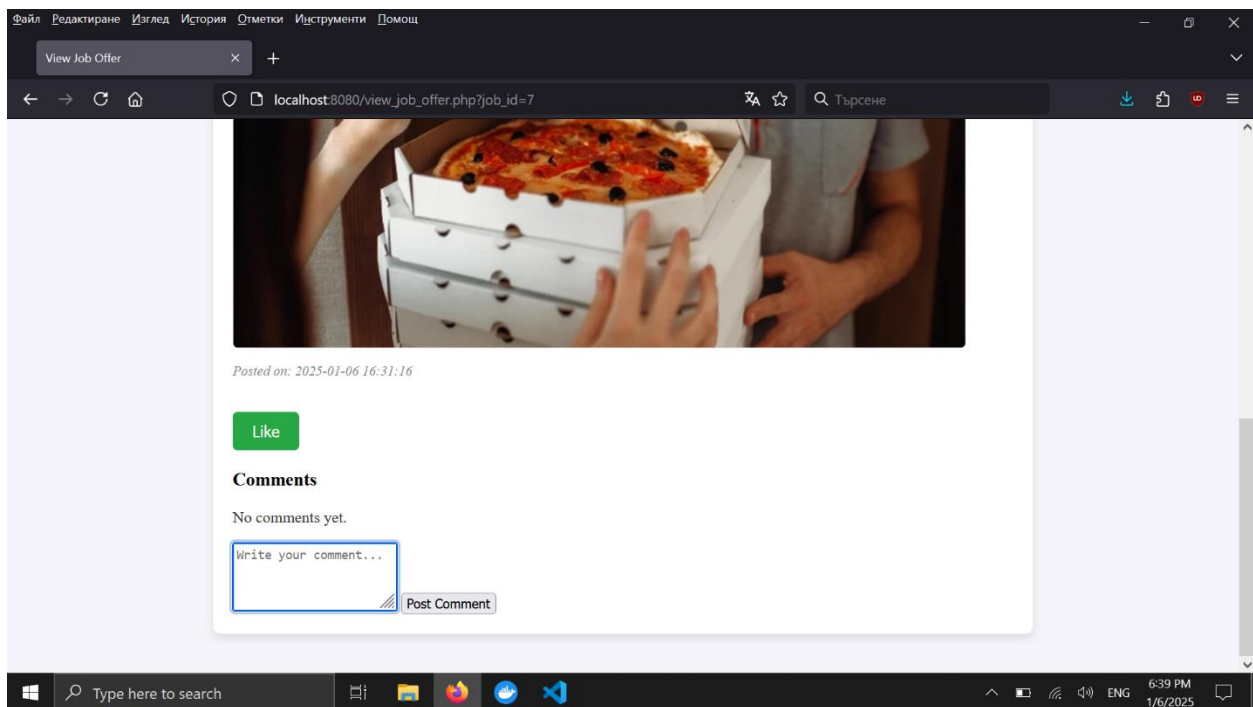
- Job Title:** A text input field.
- Job Description:** A text area.
- Phone Number:** A text input field.
- Job Images (URLs, comma separated):** A text input field with the example text 'e.g. http://image1.jpg, http://image2.jpg'.

At the bottom of the form is a green button labeled 'Submit Job Offer'.

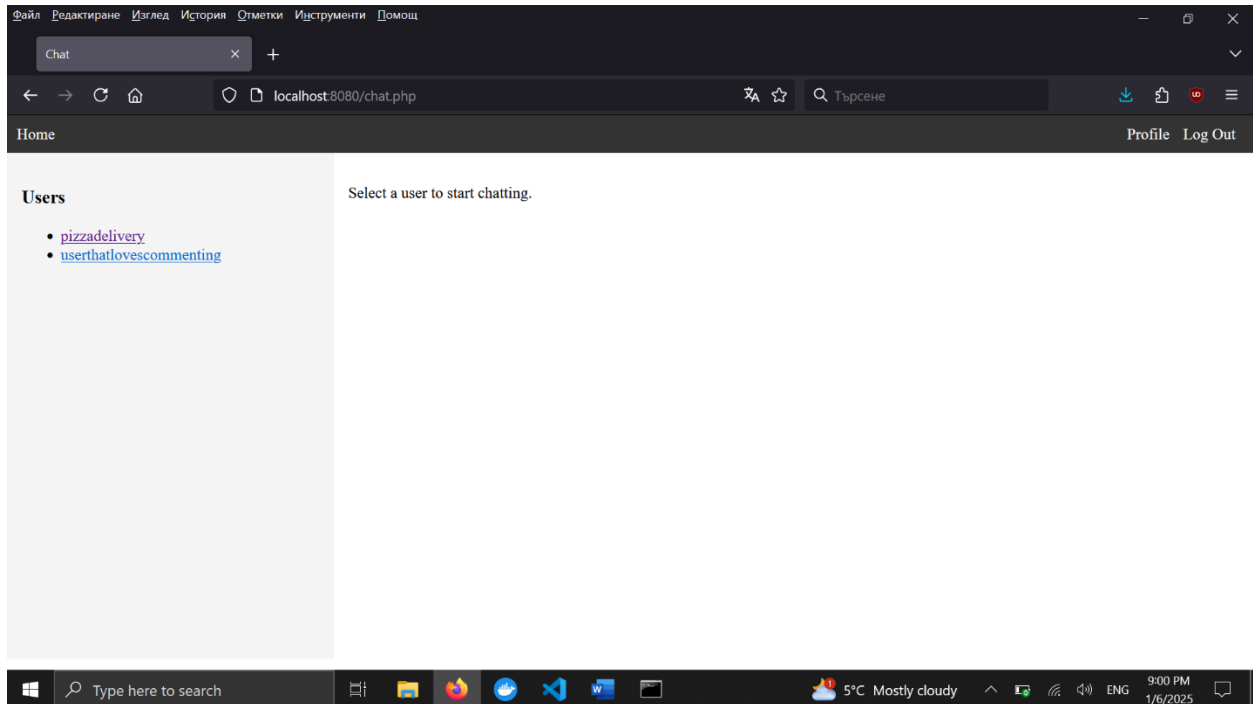
Публикуваните обяви става видими за всички, но достъпни само за оторизирани и аутентифицирани потребители.



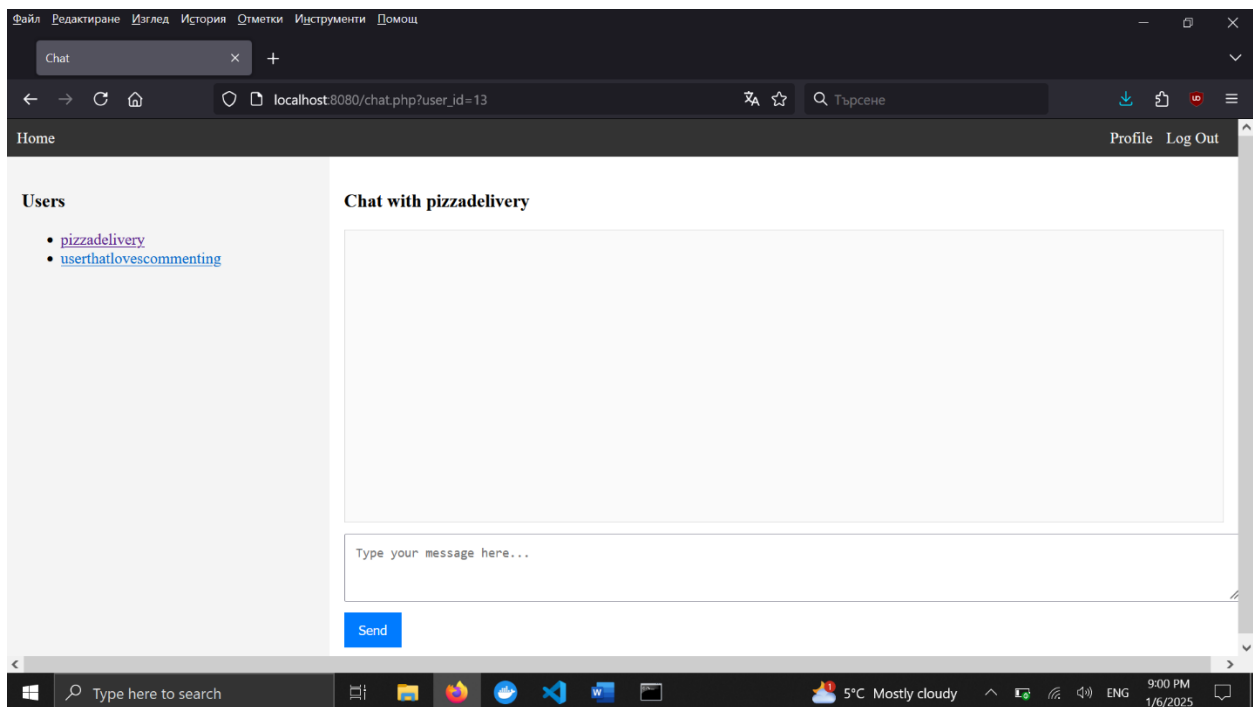
Други потребители могат да харесват, спрат да харесват или коментират създадени обяви след достъпване в тях.



Опцията за чат дава възможност потребители да общуват свободно един с друг.



От лист с всички оторизирани потребители се избира търсеният за разговор и се отваря полето за чат.



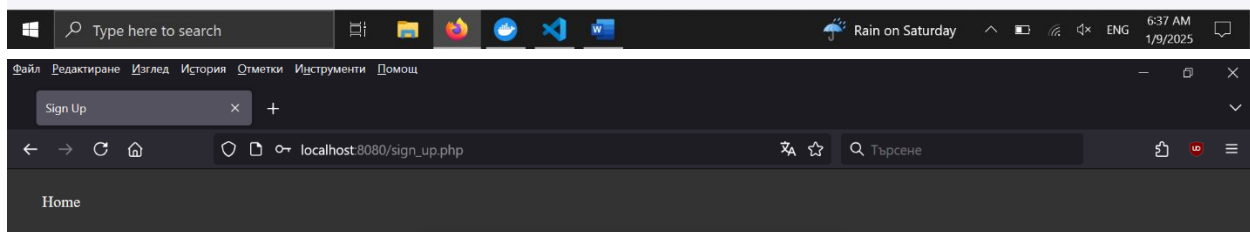
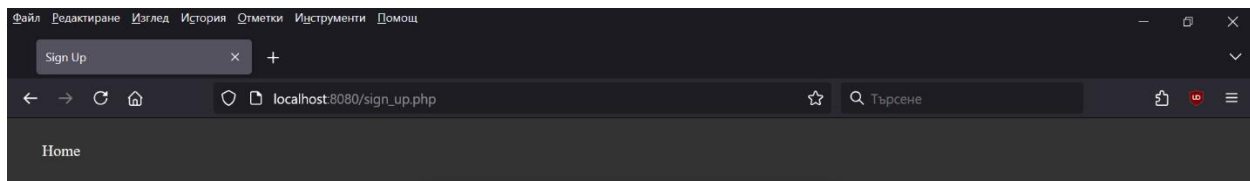
5. Приложение: Листинг на програмата с коментари, екранни форми, таблици с тестови резултати и др.

При създаването на акаунт формата задължителни параметри са името, имейла и паролата. При липса на въведени кредитеншали или грешно изписан имейл формата не се приема и се изисква съответно въвеждане на данни или поправка на имейла.

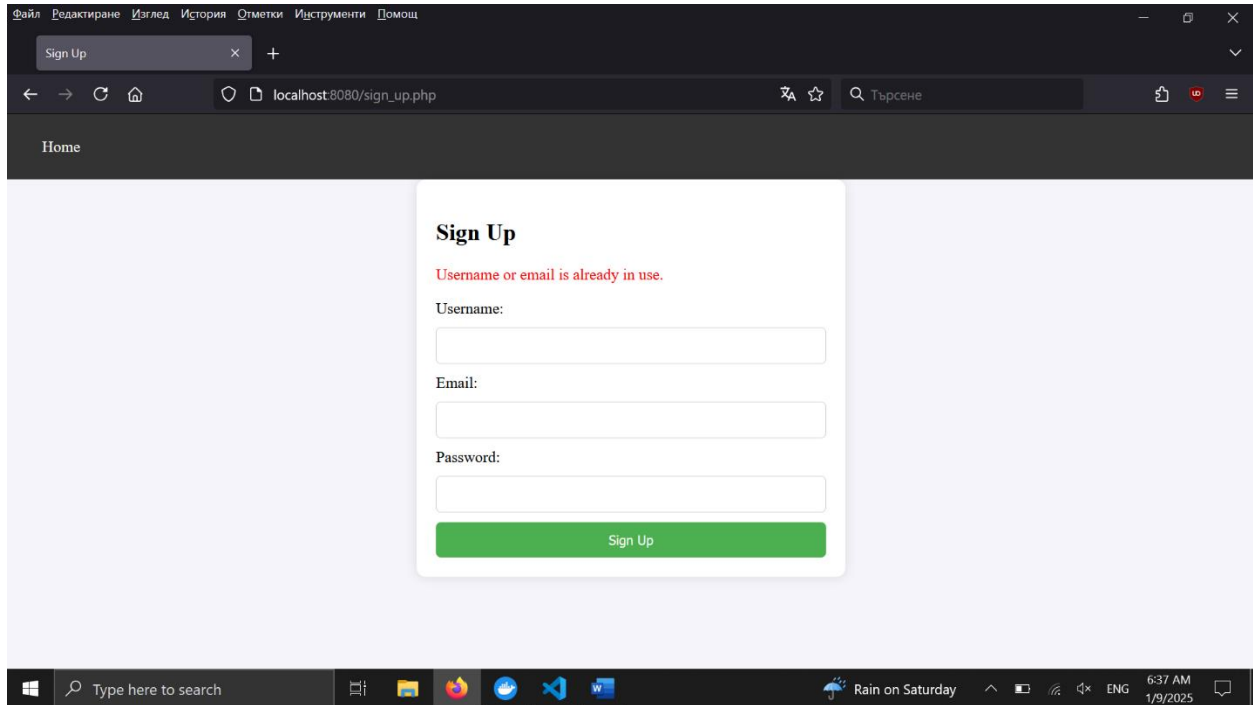
The screenshot displays a web browser window with a dark theme. The address bar shows the URL 'localhost:8080/sign_up.php'. The page content features a 'Sign Up' form with the following elements:

- Sign Up** (Section Header)
- Username:** (Label)
- (Username input field)
- (Password input field)
- Password:** (Label)
- (Password input field)
- Sign Up** (Green button)

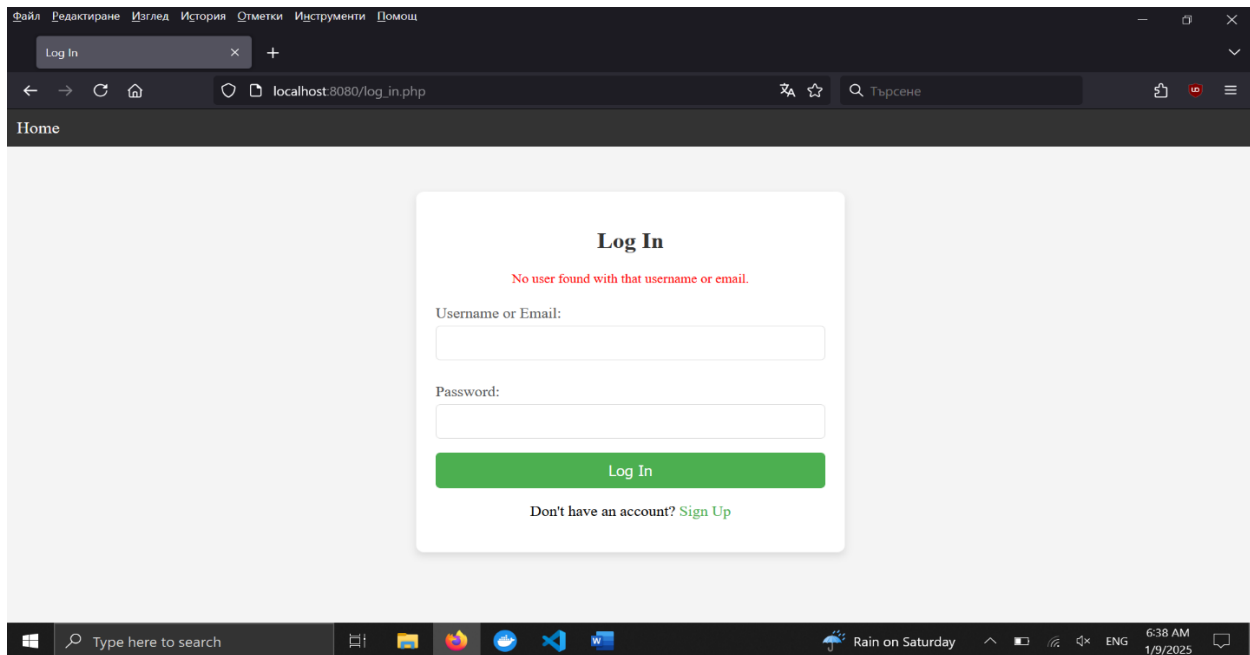
The browser's taskbar at the bottom shows the Windows logo, a search bar, and various application icons. The system tray on the right indicates the weather as 'Rain on Saturday' and the time as '6:37 AM 1/9/2025'.



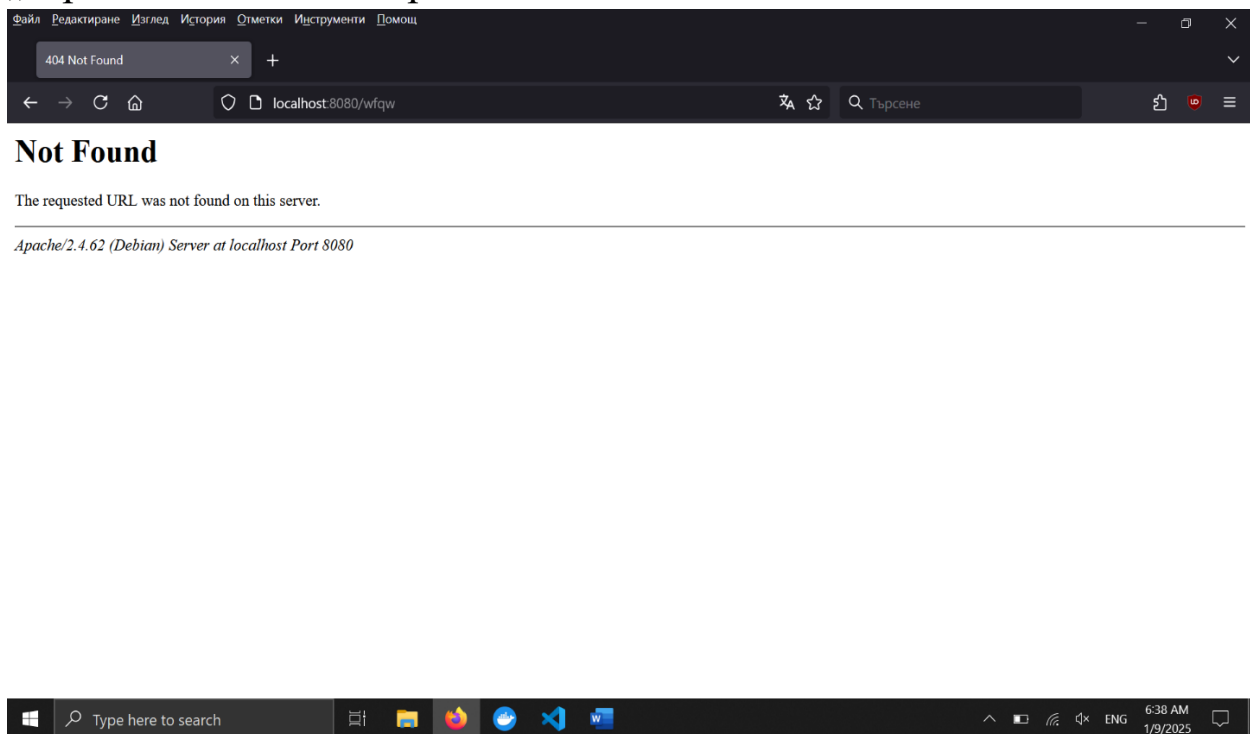
В случай, че един от credenшалите (конкретно Username или Email) са използвани преди за създаване на акаунт, ще бъде изведена грешка с червени букив “Username or email already in use.” и потребителя няма да бъде допуснат.



Във формата за логване полетата за Username/Email и Password също са задължителни. При грешно въведено потребителско име или имейл (може по избор на потребителя да въведе или потребителско име или имейл) се извежда грешка „No user found with this username or email“, защото не намира такъв съществуващ акаунт със същите въведени от потребителя credenшали.



Ако потребителя се опита да достъпи страница чрез търсачката и въведе несъществуваща страница ще получи грешка 404 „страницата не е намерена“.



При опит на потребителя да създаде оферта за работа, въвеждането на информация за обявата във всички полета е задължително.

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/create_job_offer.php'. The page title is 'Create Job Offer'. The form contains the following fields:

- Job Title:** A text input field with a tooltip that says 'Моля, попълнете полето.' (Please fill in the field).
- Text Area:** A large, empty text area for additional details.
- Phone Number:** A text input field.
- Job Images (URLs, comma separated):** A text input field with a placeholder example: 'e.g. http://image1.jpg, http://image2.jpg'.

A green button labeled 'Submit Job Offer' is located at the bottom right of the form.

6. Изпловзвана литература:

stackoverflow.com

geeksforgeeks.org