

<b>Имена:</b> Добрин Цветанов Цветков		<b>фн:</b> 81265
<b>Начална година:</b> 2015	<b>Програма:</b> бакалавър, (КН)	<b>Курс:</b> 4
<b>Тема:</b> Система за планиране и провеждане на пътуване		
<b>Дата:</b> 2019-01-23	<b>Предмет:</b> wwwTech2018_19_11ed_KN_winter	<b>имейл:</b> dobrits97@gmail.com

**преподавател:** доц. д-р Милен Петров

## ТЕМА: Система за планиране и провеждане на пътуване

### 1. Условие

Текущият проект разглежда разработването на система за планиране и провеждане на пътуване. Изискване за системата е към пътуването да може да добавяме различни точки и планирано време за минаване, както и да може да прикачваме снимки и да пишем коментари за отделните спирки, след като вече сме ги посетили.

### 2. Въведение

Избрах тази тема, защото обичам да пътувам и също така мразя да закъснявам. Винаги се стремя да си правя планове и система за планиране на пътувания би била полезна за самия мен.

На първо място реших, че трябва да можеш да виждаш само собствените си планирани пътувания и затова логично беше първата стъпка да е разработване на Login система. След приключването ѝ направих прототип на начална страница и се заех с предпазването на login/register страниците, когато вече си влязъл в системата и предпазване на началната страница, ако все още не си влязъл в системата – за целта просто се проверява дали има стартирана сесия на сървъра и ако е необходимо, потребителят бива пренасочен до приемлива страница.

Следващата стъпка беше да премина към разработването на функционалността около планирането – добавяне на нови пътувания, преглед на пътуванията, качване на коментари и накрая добавяне на снимки.

Цялата работа премина бързо и без големи затруднения.

### 3. Теория

Проектът е разработен като клиент-сървърна архитектура. За да се спести допълнителна папка в URL при достъпване от браузъра, клиентската логика не е сложена в папка client, а направо в главната директория на проекта – това са папки login, register, myTrips и newTrip, всяка от които съдържа HTML, CSS и JavaScript файлове за съответната страница, а освен това има и папка utils, която съдържа предимно JavaScript файлове, които се ползват от няколко места.

Цялата сървърна логика е изнесена в папка server, като отново имаме разделение по папки според функционалности – login, logout, register, myTrips и newTrip съдържат php файлове със сървърната логика за съответната функционалност. Освен това имаме папка images, където да се запазват изображенията, качени през нашият уебсайт. По този начин в базата данни пазим само името на файл. Последната папка отново се казва utils и съдържа неща, които се ползват на няколко места в сървърните функционалности – класове за връзки с

отделните таблици в базата данни, функция за установяване на PDO връзка с базата и функция за преобразуване на файл до base64 стринг, използвана за пращане на изображенията обратно към клиента.

#### 4. Използвани технологии

Проектът е разработен и тестван със следните технологии:

- HTML 5
- CSS 3
- JavaScript
- PHP 7.2.10
- Apache 2
- MariaDB 10.1.36
- XAMPP 7.2.10-0

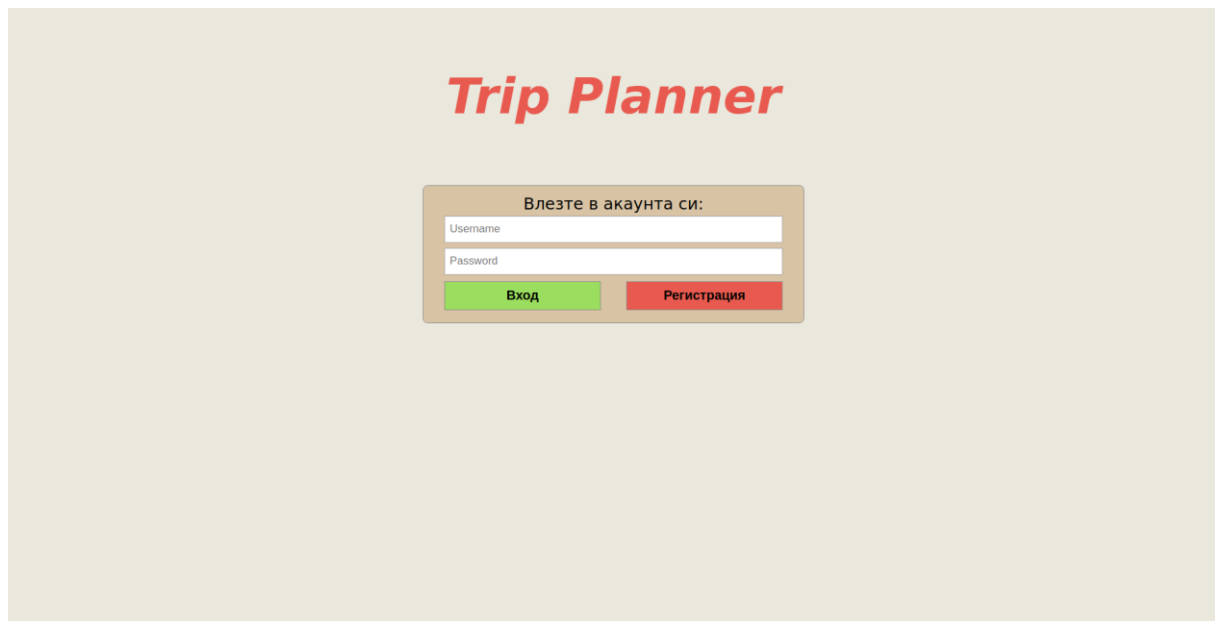
#### 5. Инсталация и настройки

За инсталация е необходимо съдържанието на архива – папка 81265\_trips - да се сложи в папката htdocs в инсталационната директория на XAMPP. След това влизаме в phpmyadmin, отиваме на таб Import и избираме файлът 81265tripsdb.sql. Този файл ще създаде база данни 81265tripsdb и след това ще създаде таблици users, trips и tripstops, като освен това ще сложи и примерни данни в тях.

За настройки, ако искаме например да променим името на базата данни, то след като използваме вградената функционалност на phpmyadmin, трябва да укажем и на приложението новото име на базата, към която да се обръща. Това можем да направим във файлът conf.php, който се намира в папка server. Този файл указва различни настройки за приложението като име на базата данни, адрес, на който е достъпна, акаунт за достъп и т.н.

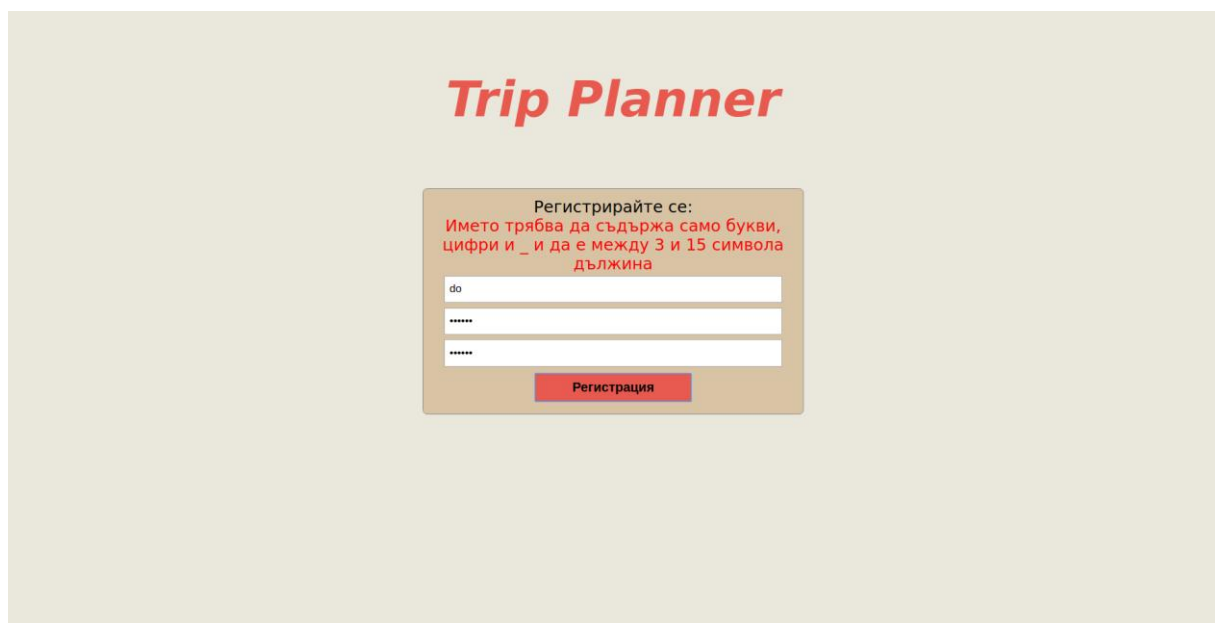
#### 6. Кратко ръководство на потребителя

За да пуснем приложението след като сме сложили папка 81265\_trips в папката htdocs, отиваме на адрес „localhost/81265\_trips“. В главната директория има файл index.php, който ще се изпълни при достъп до този адрес и при наличие на сесия ще ни изпрати на адрес „localhost/81265\_trips/myTrips/“, а ако такава няма ще бъдем препратени на адрес „localhost/81265\_trips/login“, който изглежда по следния начин:



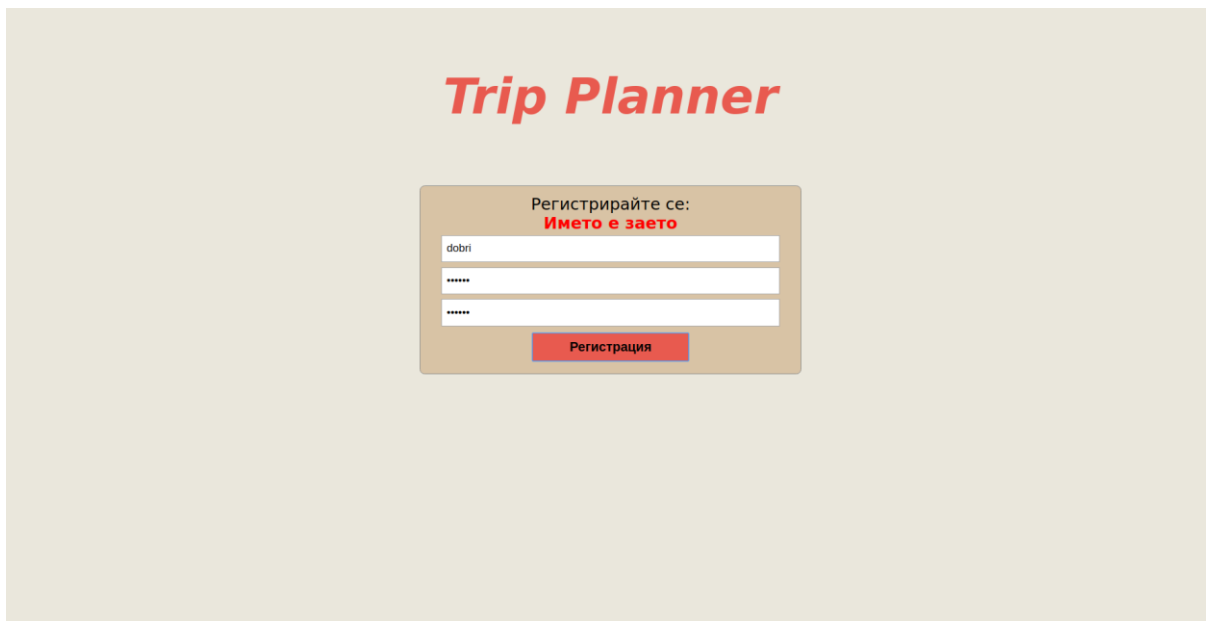
Фиг1. Страница login

При първо посещение на сайта цъкаме бутонът register и биваме препратени на съответната страница:



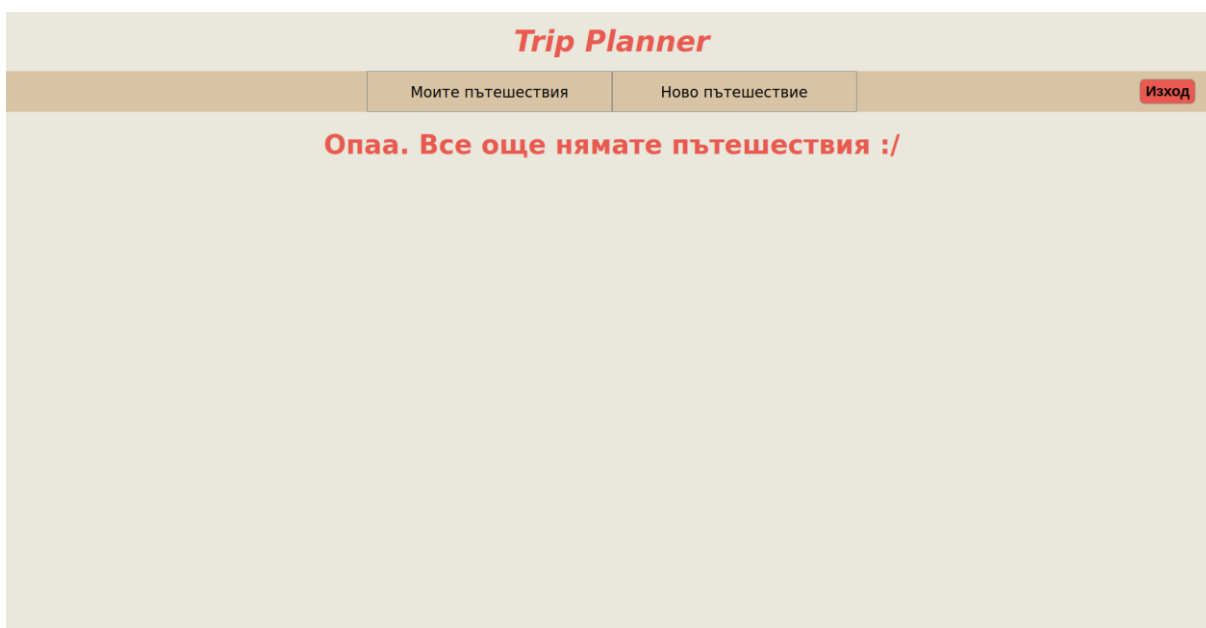
Фиг2. Страница register

Както се вижда, имаме полета за потребителско име и 2 полета за парола, които трябва да съвпадат. Има валидация както от клиентската страна, така и от сървърната страна:



Фиг3. Страница register – сървърна валидация

След успешен вход/регистрация биваме препратени на страницата myTrips. Ако акаунтът е току що създаден, то все още няма да имаме пътешествия и ще видим следният екран:



Фиг4. Страница myTrips, когато все още нямаме създадени пътешествия

Натискаме на „Ново пътешествие“ и ще видим форма за създаване на пътешествие с поле за име на пътешествието, бутон за динамично добавяне на още спирки и бутон за създаване на пътешествието:

Фиг5. Страница newTrip

След успешно създаване биваме препратени обратно на страницата myTrips, където вече имаме падащо меню със създадените пътешествия. Избираме някое, за да видим подробности:

Фиг6. Преглед на създадено пътешествие

Тук вече имаме възможност за всяка спирка да добавим изображение, което автоматично се запазва на сървъра. Освен това можем да напишем коментари по спирките и да ги запазим чрез натискането на бутон „Запази коментарите“.

Освен това, по всяко време в горният десен ъгъл имаме бутон „Изход“, който ще развали сесията и ще ни прати обратно на login екранът.

## 7. Примерни данни

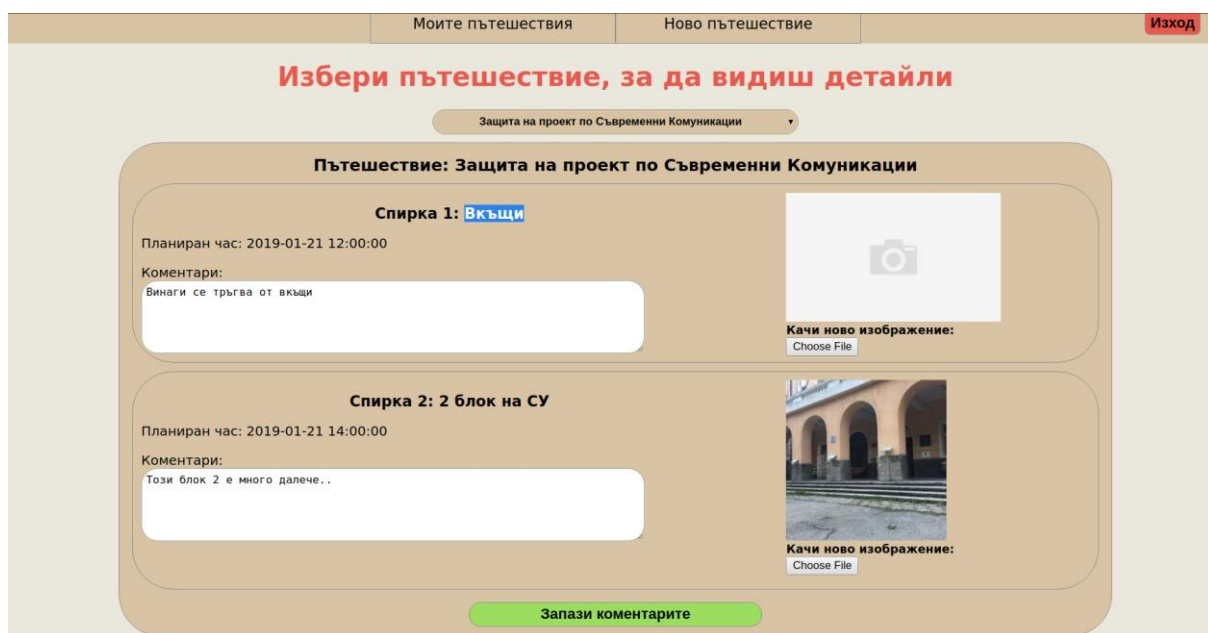
Приложението е тествано и качено със следните данни:

- Потребител web без никакви пътешествия
- Потребител dobri с две пътешествия:
  - Пътешествие „Вкъщи – ФМИ“ с две спирки:
    - Спирка 1: Вкъщи
    - Спирка 2: ФМИ

The screenshot shows a web interface for managing travel itineraries. At the top, there are tabs for 'Моите пътешествия' (My Itineraries) and 'Ново пътешествие' (New Itinerary), with an 'Изход' (Logout) button on the right. Below the tabs, a red heading reads 'Избери пътешествие, за да видиш детайли' (Select an itinerary to see details). A dropdown menu shows 'Вкъщи - ФМИ' selected. The main content area displays details for this itinerary, including two stops: 'Спирка 1: Вкъщи' and 'Спирка 2: ФМИ'. Each stop has a planned time, a comment field, and an image upload section. The first stop is planned for 2019-01-23 08:00:00 with the comment 'Трудно ще е ставането, но няма как.' and a placeholder image. The second stop is planned for 2019-01-23 09:00:00 with the comment 'Дано защитата на проекта мине успешно!' and another placeholder image. A green button at the bottom says 'Запази коментарите' (Save comments).

Фиг7. Пътешествие „Вкъщи - ФМИ“

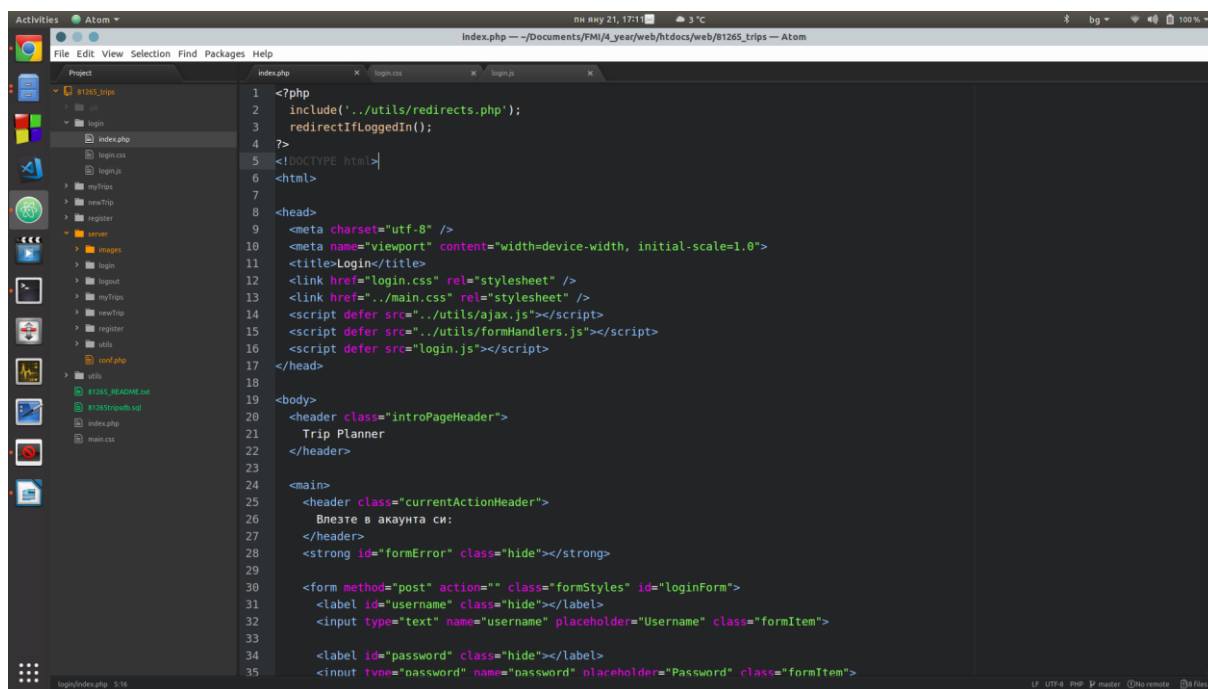
- Пътешествие „Защита на проект по Съвременни Комуникации“, отново с две спирки
  - Спирка 1: Вкъщи
  - Спирка 2: 2 блок на СУ



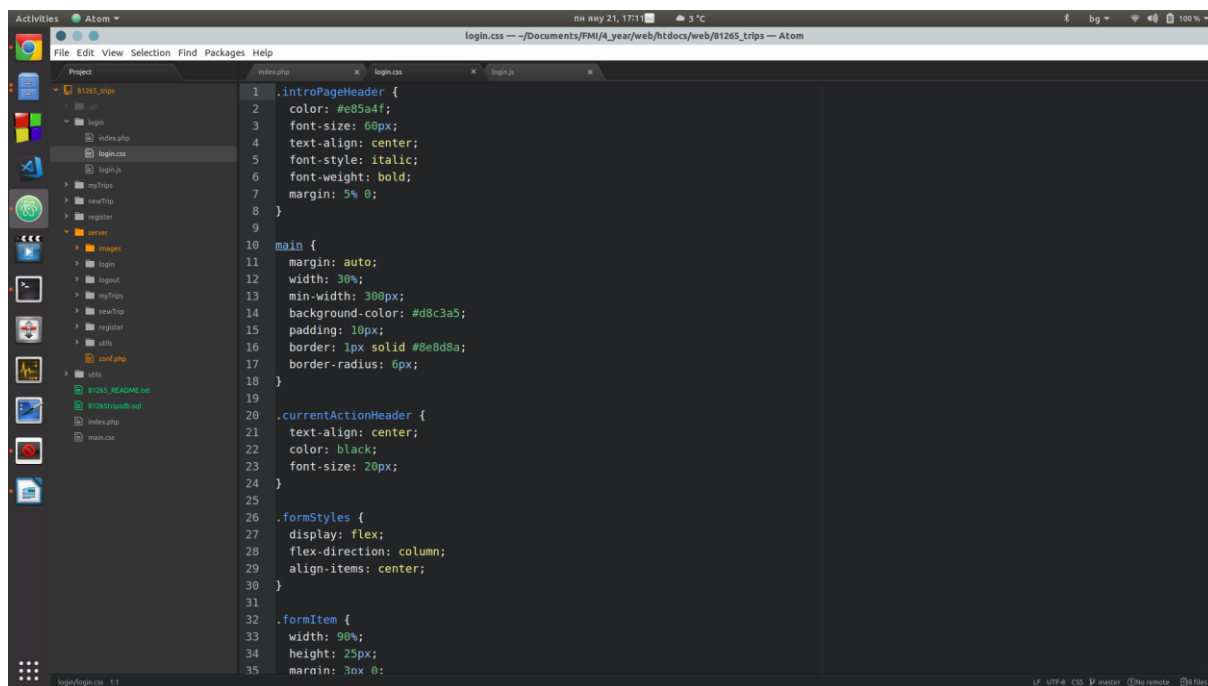
Фиг8. Пътешествие „Защита на проект по Съвременни Комуникации“

## 8. Описание на програмния код

Кодът е разделен, така че да е интуитивно кое как се случва. В клиентската част разделението е по следния начин: статичното съдържание на страницата е в html(или php) файл, стиловете са в css файл, динамичното генериране на html и event handlers са в javascript файловете. Например при login екранът кодът изглежда така:

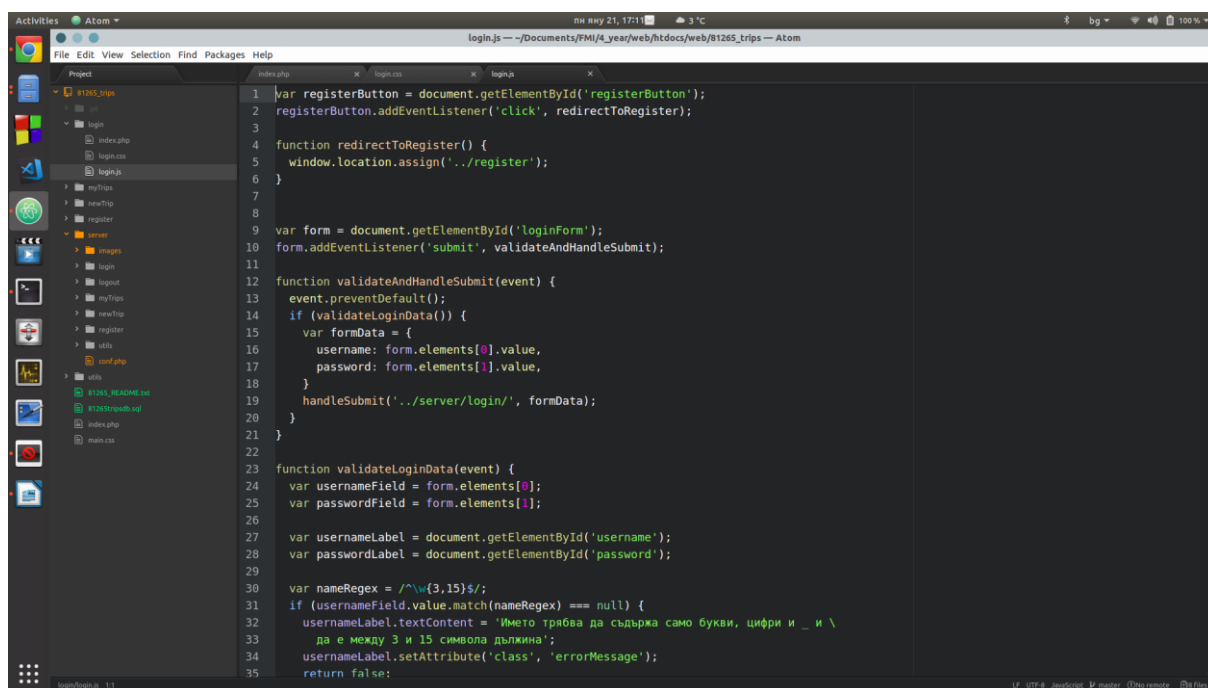


Фиг9. Файл login/index.php



```
1 .introPageHeader {
2   color: #e85a4f;
3   font-size: 60px;
4   text-align: center;
5   font-style: italic;
6   font-weight: bold;
7   margin: 5% 0;
8 }
9
10 .main {
11   margin: auto;
12   width: 30%;
13   min-width: 300px;
14   background-color: #d8c3a5;
15   padding: 10px;
16   border: 1px solid #8e8d8a;
17   border-radius: 6px;
18 }
19
20 .currentActionHeader {
21   text-align: center;
22   color: black;
23   font-size: 20px;
24 }
25
26 .formStyles {
27   display: flex;
28   flex-direction: column;
29   align-items: center;
30 }
31
32 .formItem {
33   width: 90%;
34   height: 25px;
35   margin: 3px 0;
```

Фиг10. Файл login/login.css



```
1 var registerButton = document.getElementById('registerButton');
2 registerButton.addEventListener('click', redirectToRegister);
3
4 function redirectToRegister() {
5   window.location.assign('../register');
6 }
7
8
9 var form = document.getElementById('loginForm');
10 form.addEventListener('submit', validateAndHandleSubmit);
11
12 function validateAndHandleSubmit(event) {
13   event.preventDefault();
14   if (validateLoginData()) {
15     var formData = {
16       username: form.elements[0].value,
17       password: form.elements[1].value,
18     }
19     handleSubmit('../server/login/', formData);
20   }
21 }
22
23 function validateLoginData(event) {
24   var usernameField = form.elements[0];
25   var passwordField = form.elements[1];
26
27   var usernameLabel = document.getElementById('username');
28   var passwordLabel = document.getElementById('password');
29
30   var nameRegex = /^\\w(3,15)$/;
31   if (usernameField.value.match(nameRegex) === null) {
32     usernameLabel.textContent = 'Името трябва да съдържа само букви, цифри и _ и \\
33     да е между 3 и 15 символа дължина';
34     usernameLabel.setAttribute('class', 'errorMessage');
35     return false;
```

Фиг11. Файл login/login.js

На сървърната страна имаме само php файлове, като повечето от тях отговарят за обработка на заявки, а една малка част за връзка с базата данни:



```

1 <?php
2 include('../utils/TripDbConnector.php');
3
4 if ($_SERVER['REQUEST_METHOD'] === 'POST') {
5     $rawData = file_get_contents('php://input');
6     $dataArray = json_decode($rawData, true);
7
8     if (invalidTripName($dataArray['tripName'])) {
9         header('HTTP/1.1 400 Invalid trip name');
10        echo 'Невалидно име на пътешествието';
11        return;
12    }
13
14    $tripDb = new TripDbConnector();
15    if ($tripDb->tripNameTaken($dataArray['tripName'])) {
16        header('HTTP/1.1 409 Trip name taken');
17        echo 'Името е заето';
18        return;
19    }
20
21    $trip = $tripDb->newTrip($dataArray);
22    echo "Great Success";
23 }
24
25 function invalidTripName($tripName)
26 {
27     return $tripName === "";
28 }
29
30

```

Фиг12. Файл server/newTrip/index.php

```

1 <?php
2 include('dbUtils.php');
3 include('TripStopsDbConnector.php');
4 if (session_status() !== PHP_SESSION_ACTIVE) {
5     session_start();
6 }
7
8 class TripDbConnector
9 {
10     private $conn;
11
12     public function __construct()
13     {
14         $this->conn = createPDO();
15     }
16
17     public function __destruct()
18     {
19         $this->conn = null;
20     }
21
22     public function tripNameTaken($tripName)
23     {
24         $trip = $this->getTrip($tripName);
25         return boolval($trip);
26     }
27
28     public function getTrip($tripName)
29     {
30         $getTripStatement = $this->conn->prepare("SELECT * FROM Trips
31         WHERE ownerId = ? AND name = ?");
32         $getTripStatement->execute([$_SESSION['userId'], $tripName]);
33         return $getTripStatement->fetch();
34     }
35 }

```

Фиг13. Файл server/utils/TripDbConnector.php

## 9. Приноси на студента, ограничения и възможности за бъдещо разширение

Една възможност за разширение в бъдещето е добавяне на географска ширина и дължина на спирките и визуализирането им чрез Google Maps. За съжаление тази функционалност не бе реализирана с текущата версия, поради незапознатост с Google API-то, таксата за използване на Google Maps на собствен сайт и малкото оставащо време до крайният срок за проектите след като получих тази идея от преподавателя.

Друга възможност за разширение е препоръчване на спирки – на база на спирките, които си избрал до момента, приложението да ти каже другите потребители какво още

посещават след тези спирки. За целта ще трябва да се използват знания както по УЕБ технологии, така и по Изкуствен интелект.

## 10. Какво научих

На първо място научих как да използвам PHP за писане на сървърна логика. Това включва работа със сесии, обработка на заявки, как да променя HTTP response code-а на отговора, представяне на изображенията като base64 стринг, връзка с базата данни и т.н.

Освен това проектът ми помогна да обогатя знанията си по front-end частта. Затвърдих знанията си по това как да закачам eventHandler-и и да генерирам динамично HTML, използвайки само JavaScript без framework. Срещнах някои нови препятствия при писането на CSS, но успях да се справя.

## 11. Използвани източници

- <https://developer.mozilla.org/en-US/>
- <https://www.w3schools.com/>
- <https://stackoverflow.com/>

Предал (подпис): .....

*/фн, имена, спец., група/*

Приел (подпис): .....