CSCI 4131 – Internet Programming
Homework Assignment 6

Posted March 25th, 2018

Due: Friday April 6 at 11:55 PM

Submissions Accepted (with Penalty) through Saturday April 7 at 11:55AM

1 Description

Assignment 5 provided an introduction to web-development using Node.js. This assignment will build upon the what you have developed with assignment 5. The objective of this assignment is to develop a basic website using Express. Express is an application framework that simplifies the development of node.js sever-side applications, and it is the most widely used application framework for doing so. Typical features of Express are:

- routing: a way to map urls and http verbs to code paths
- easy methods for parsing http requests and building http responses:

*The following are **some of the resources** you should use to familiarize yourself with Express:*

- Essential
  - Installing Express
  - Hello world example of Express
  - Basic routing in Express
  - Serving static files in Express
- Extra
  - Express website
  - Books and blogs
  - FAQ
  - Routing in Express
  - API Reference
  - Building a Website with Node.js and Express.js (Video tutorial)

This assignment will also introduce you to SQL and the MySQL database.

The following are resources you should review to get familiar with SQL, MYSQL and MYSQL/Node.js

- https://www.w3schools.com/sql/
- https://www.w3schools.com/sql/sql_ref_mysql.asp
- https://www.w3schools.com/nodejs/nodejs_mysql.asp
- PHP and MYSQL: Chapter 17 pp.542-565; Chapter 18

2 Preparation and Provided Files

I. The first step will be to get Node.js and MySQL running on CSE lab machines. This can be accomplished as follows:

1. Log into a CSE lab machine.
2. Most of the CSE lab machines run version 8.9.4 of Node.js (as of 03/23/2018) which is the most current version. Vole runs version 6.12.3 which will also be sufficient for this assignment.
3. Open the terminal and type the following command to add the Node.js module:

   ```
   module add soft/nodejs
   ```

4. The next step is to check the availability of Node.js. Type the following command to check the version of Node.js on the machine:

   ```
   node -v
   ```

5. This will display the current installed version.
6. To user the MYSQL database, you will need a database user id and password. Your MYSQL database user id and numeric password can be found on the class Moodle site in the grade item named **"DB Credentials"**. The feedback section of this grade item has the database user name and password. The user name and password are separated by a space.
7. At the terminal, type the following command to add MySQL module:

   ```
   module add soft/mysql
   ```

8. At the terminal, type the following command to login to MySQL and check whether its active:

   ```
   mysql -u your_database_user -hcse-curly.cse.umn.edu -P3306 -p your_database_user
   ```

   Replace your_database_user with the database id provided to you before hitting enter. your_database_user will be in the format: C4131S18UXXX

   When prompted for password, enter the numeric password provided to you.
9. After successful login, you should see **mysql>** prompt.

II. The second step is to create a Node.js (Express) project for this assignment. This can be accomplished as follows:

1. Open the terminal on a CSE lab machine.
2. Create a directory named <x500id_hw05> by typing the following command:

   ```
   mkdir yourx500id_hw06
   ```

3. Go inside the directory by typing the following command:

   ```
   cd  yourx500id_hw06
   ```

4. Having a file named *package.json* in Node.js project makes it easy to manage module dependencies and makes the build process easier. To create *package.json* file, type the following command:

   ```
   npm init
   ```

5. The *npm init* command will prompt you to enter the information. Use the following guidelines to enter the information (The things that you need to enter are in **bold** font. Some fields can be left blank.):

```
name: (yourx500id_hw08) yourx500id_hw06

version: (1.0.0) <Leave blank>

description: Assignment 6

entry point: (index.js) <Leave blank> (You will create an index.js
file for your use)

test command: <Leave blank>

git repository: <Leave blank>

keywords: <Leave blank>

author: yourx500id

license: (ISC) <Leave blank>
```

6. After filling in the above information, you will be prompted to answer the question: "`Is this ok? (yes)`". Type **yes** and hit enter.

7. Listing all the available files in the directory should display the following:

```
drwx------  4 nayan003 CSEL-student    7 Mar  3 03:50 ./

drwx------ 12 nayan003 CSEL-student   17 Mar  3 03:50 ../

-rwxrwxrwx  1 nayan003 CSEL-student  309 Mar  3 03:37 package.json
```

8. Install Express by typing the following command:

```
npm install --save express
```

9. You can use any npm module that you deem fit for this assignment. Some npm modules that might be useful for this assignment are and should be installed are:

- mysql (*npm install --save mysql*)
- body-parser (*npm install --save body-parser*)
- express-session (*npm install --save express-session*)
- sha1 (*npm install --save sha1*)

10. You are free to decide your own project structure for this assignment.

NOTE: We have provided a sample file for server (sample_index.js) which can be used for reference.

III. Database setup:

1. The following files have been provided to you for this assignment:
   - create_accounts_table.js
   - insert_into_accounts_table.js
   - create_places_table.js

2. Download these files and move them to `yourx500id_hw06` directory.
3. Edit these files as per the instructions provided inside these files.
4. At the terminal, type the following command to create the MySQL table: tbl_accounts

   ```
   node create_accounts_table.js
   ```

   This table will be used to store your encrypted login credentials.

5. At the terminal, type the following command to insert values for acc_name, acc_login, acc_password into tbl_accounts table:

   ```
   node insert_into_accounts_table.js
   ```

   You will use the values chosen for acc_login and acc_passowrd to login to the website. Keep the values in a safe place and do not share them with anyone

6. At the terminal, type the following command to create the MySQL table: tbl_places

   ```
   node create_places_table.js
   ```

   This table will be used to store the details of places.

At this point you are ready to start the website development.

3 Functionality

Your website will have 3 pages:

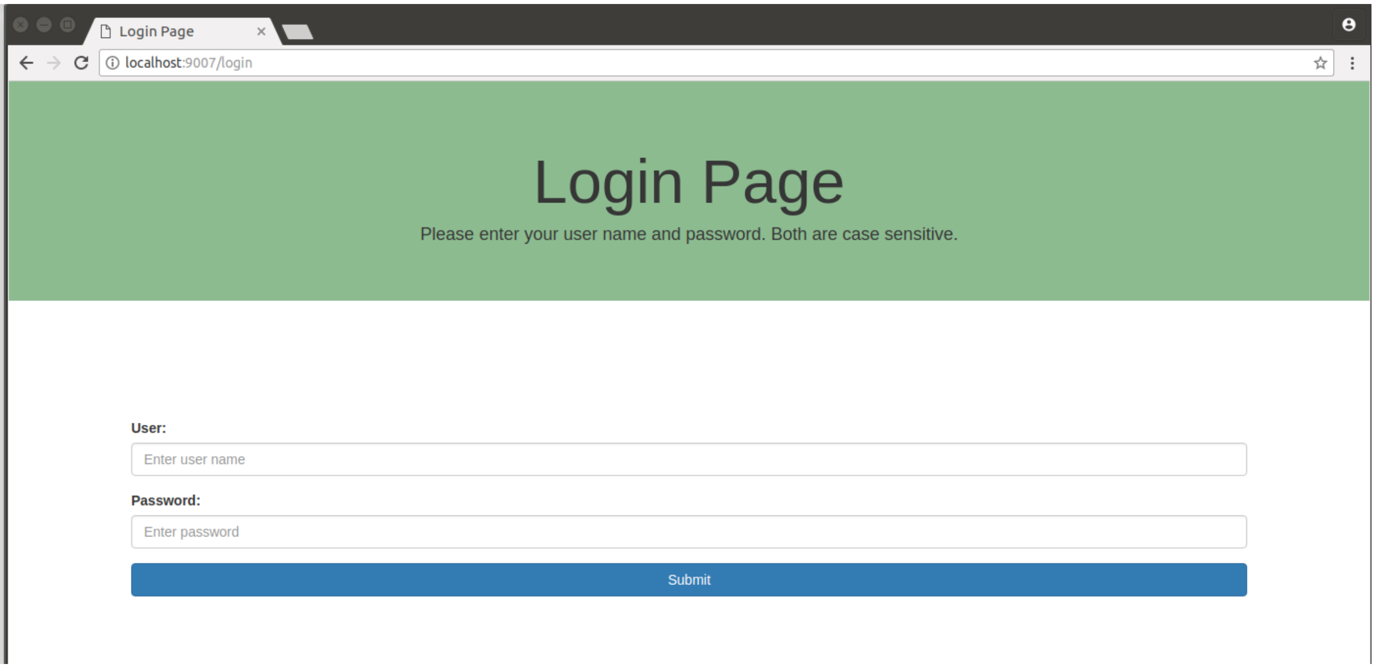
- A **Login** page
- A **Favourite places** page
- An **Add place** page

The *Favourite places* and *Add place* pages will have a navigation bar with logout button.

NOTE: You will need to develop the entire website including frontend (HTML pages, CSS, Javascript) and backend (Express server).

Follow the instructions provided below:

Login page



Login Page

Please enter your user name and password. Both are case sensitive.

User:

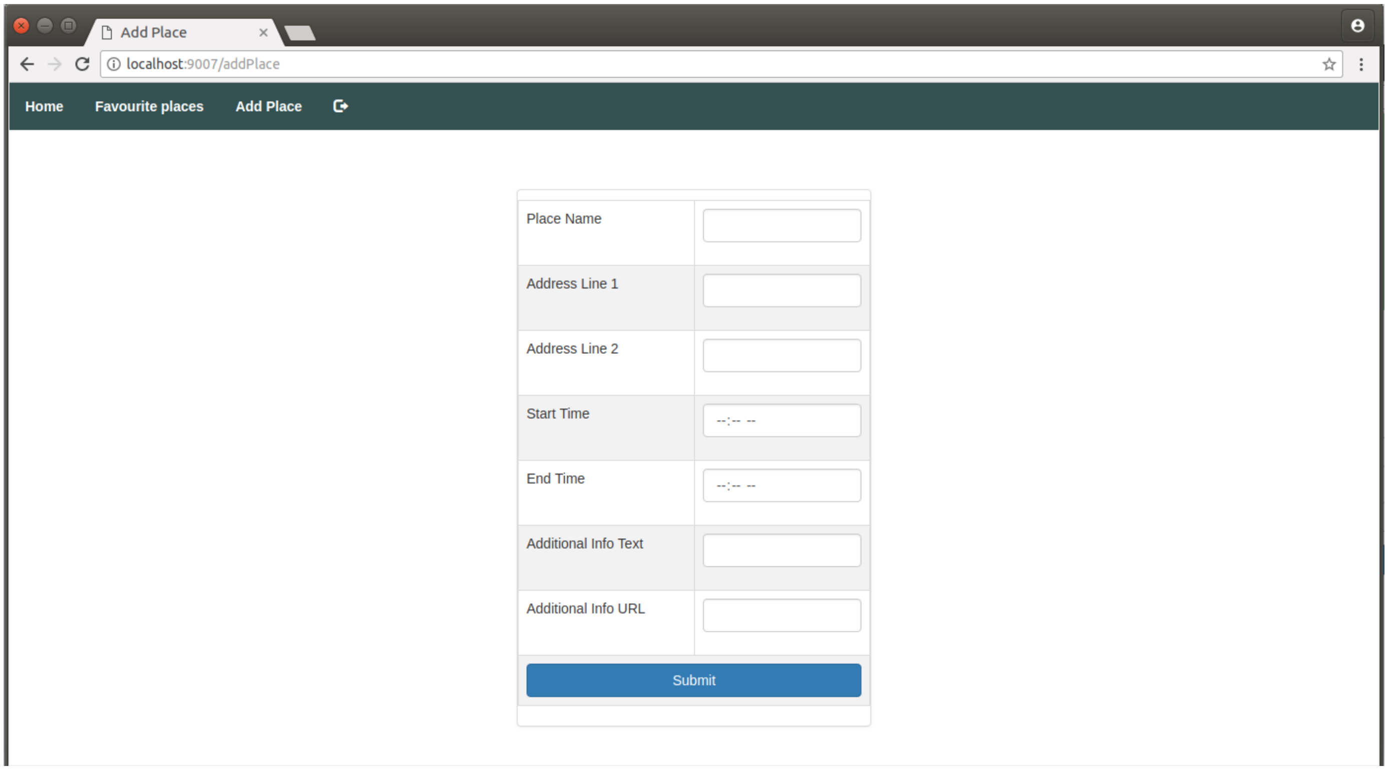
Enter user name

Password:

Enter password

- This page should have a form with two fields: “User”, and “Password”
- Both these fields are mandatory.
- When the submit button is clicked, the values entered for “User” and “Password” should be sent to the server for validation before allowing further access to website.
- The server will validate the received values against the *acc_login*, and *acc_password* fields of *tbl_accounts*. Passwords is stored in the database in a SHA1 hashed format in tbl_accounts. Server should convert the received password string into a SHA1 hash format and compare it to the SHA1 hashed password stored in the database. (*Hint: you can use sha1 npm module*)
- Upon successful validation, server should
  - Create a user session (*Hint: you can use express-session module*).
  - Send a response back to client indicating successful validation.
- If the validation fails, server should:
  - Send a response back to client indicating validation failure.
- If successful response is received from server, user should be routed to “Favourite places” page, otherwise appropriate error message should be displayed to the user (Check screenshots towards the end of this assignment)

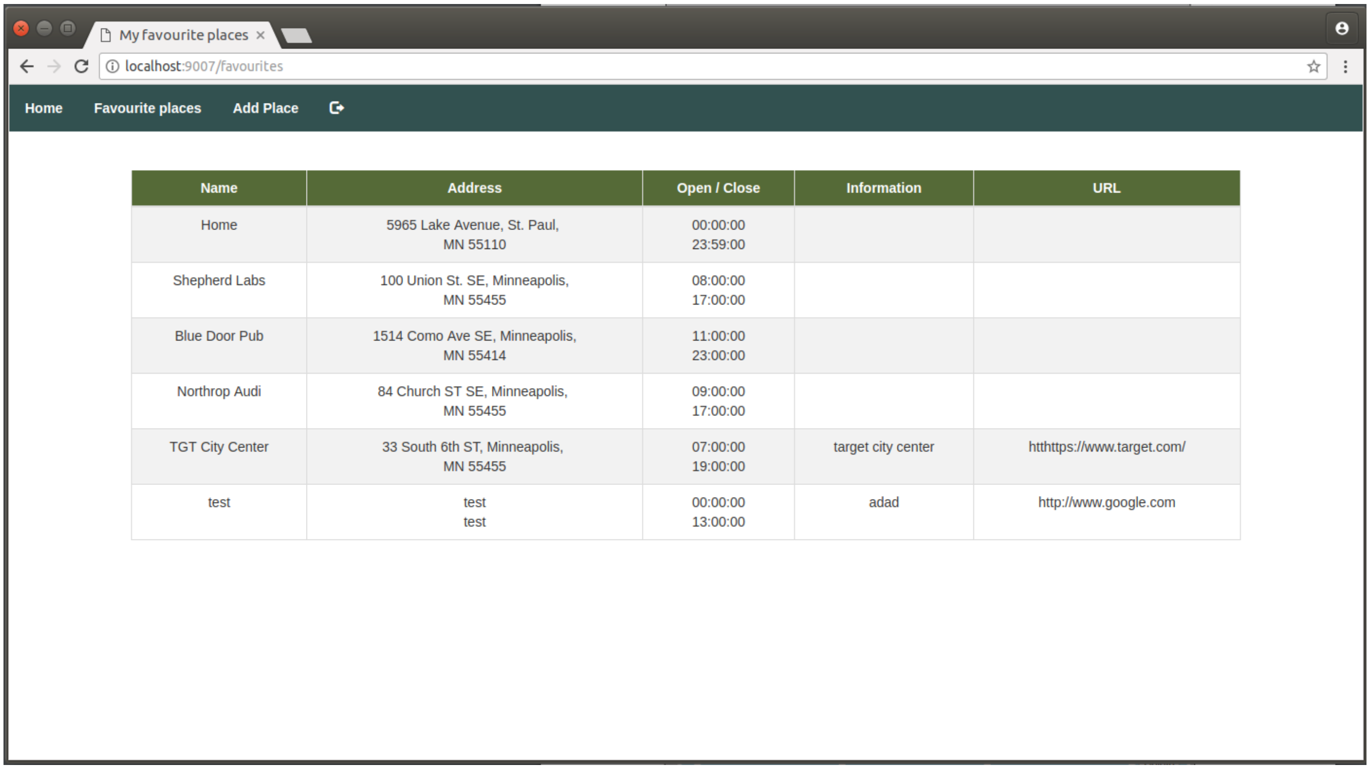
Add Place page



Place Name	
Address Line 1	
Address Line 2	
Start Time	--:-- --
End Time	--:-- --
Additional Info Text	
Additional Info URL	

- You can use the form developed in the earlier assignments for ‘Add Place’ page.
- If this page is accessed without a valid login, the user should be routed to “**Login**” page.
- The page should have a navigation bar with logout button.
- All the form validations from earlier assignments should be retained:
  - Place Name, Address Line 1, Start Time, End Time are mandatory,
  - Place Name and Address should accept only alphanumeric characters and spaces.
- Upon clicking submit, the form data should be posted to the server.
- The server should insert the received data into the following table: **tbl_places** (*Hint: you can use mysql module*)
- The mapping between form fields and table columns is:
  - Place Name: **place_name**
  - Address Line 1: **addr_line1**
  - Address Line 2: **addr_line2**
  - Start Time: **open_time**
  - End Time: **close_time**
  - Additional Info Text: **add_info**
  - Additional Info URL: **add_info_url**
- Upon successful insertion, server should route the user to the “Favourite places” page.

Favorite places page



Name	Address	Open / Close	Information	URL
Home	5965 Lake Avenue, St. Paul, MN 55110	00:00:00 23:59:00		
Shepherd Labs	100 Union St. SE, Minneapolis, MN 55455	08:00:00 17:00:00		
Blue Door Pub	1514 Como Ave SE, Minneapolis, MN 55414	11:00:00 23:00:00		
Northrop Audi	84 Church ST SE, Minneapolis, MN 55455	09:00:00 17:00:00		
TGT City Center	33 South 6th ST, Minneapolis, MN 55455	07:00:00 19:00:00	target city center	htthttps://www.target.com/
test	test test	00:00:00 13:00:00	adad	http://www.google.com

- If a user tries to access this page without a valid login, the user should be routed to "**Login**" page.
- The page should have a navigation bar with logout button.
- The table in this page should be dynamically populated.
- To achieve this, the server should provide a GET API which returns the list of favourite places. This API will be very similar to the one developed in assignment 5. It will get the list of favourite places by querying the following table: tbl_places.
- The client will call this API and populate the table using the data received from the server.

Logout button

- Upon clicking the logout button, the user session should be destroyed and user should be routed to **Login** page.

4 Submission Instructions

PLEASE ENSURE TO TEST YOUR CODE ON CSE LAB MACHINES.

You will need to submit all the files used to develop the website. This includes all the HTML, CSS, JavaScript, package.json, index.js and any other files.

Towards this end, make a copy of your working directory: `yourx500id_hw06`. Rename the copied folder as **yourx500id_express**. Delete **node_modules** folder present inside the copied directory (yourx500id_express).

Create a README file inside yourx500id_express directory. This README file should include: Your x500id, acc_login, and acc_password values from insert_into_accounts_table.js file.

Compress the **yourx500id_express** directory and submit it**.**

We will use the acc_login and acc_password values to login to your website. Ensure that these values are correct and can be used to access your website.

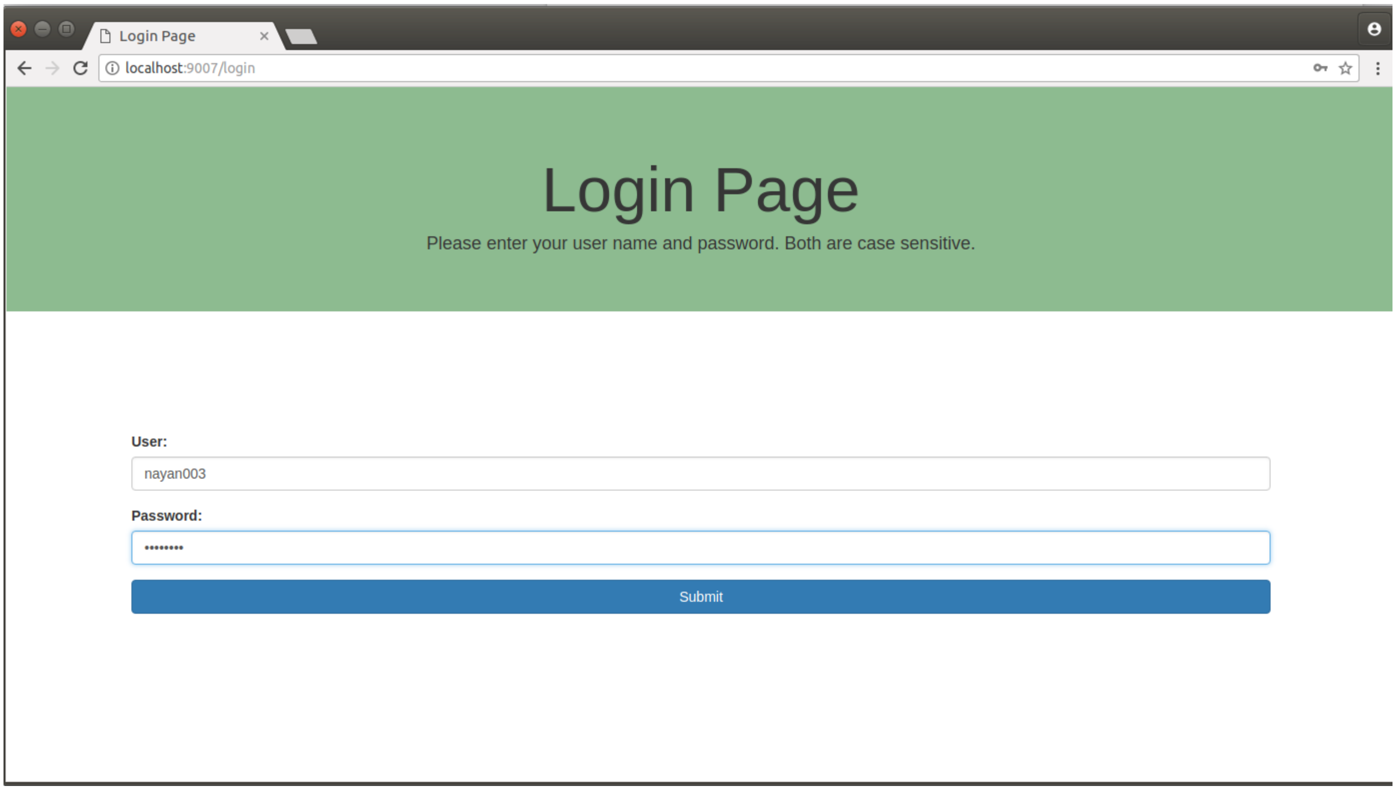
5 Evaluation

Your submission will be graded out of 100 points on the following items:

- Submission instructions are met. **(5 points)**
- The "Favourite places" and "Add Place" pages of your website redirect the user to "Login" page automatically before authentication. **(10 points)**
- The "Login" page shows the form elements and submit button. **(5 points)**
- After successful login validation by the server, the "Login" page redirects the user to "Favourite places" page. **(10 points)**
- If server login validation fails, an error message is displayed on "Login" page and user is not allowed further access. **(10 points)**
- The "Favourite places" page displays correctly and has operational navigation bar. **(5 points)**
- The "Favourite places" page gets the list of favorite places from server. The favorite places are dynamically added to the table displayed in the user's browser. **(20 points)**
- The "Add Place" page displays and has an operational navigation bar. **(5 points)**
- The user can add a new place using the form present in "Add place" page. **(15 points)**
- After a new place is added using the "Add Place" page, the user is redirected to "Favourite places" page, and the users favorite places are correctly displayed. **(5 points)**
- The logout functionality works correctly. **(10 points)**

6 Screenshots (See the following pages for examples)

Login Page



Login Page

localhost:9007/login

Login Page

Please enter your user name and password. Both are case sensitive.

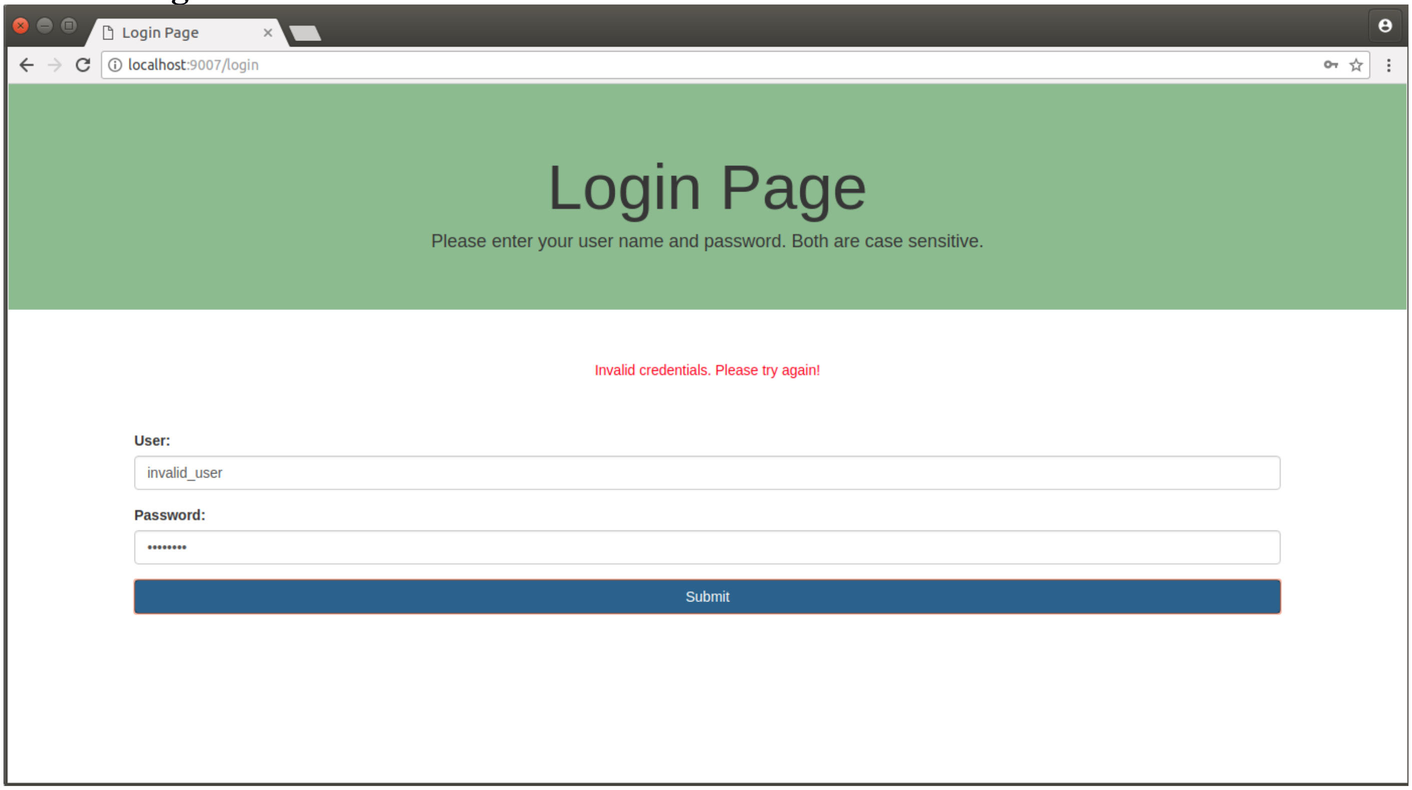
User:

nayan003

Password:

••••••••

Invalid Login



Login Page

localhost:9007/login

Login Page

Please enter your user name and password. Both are case sensitive.

Invalid credentials. Please try again!

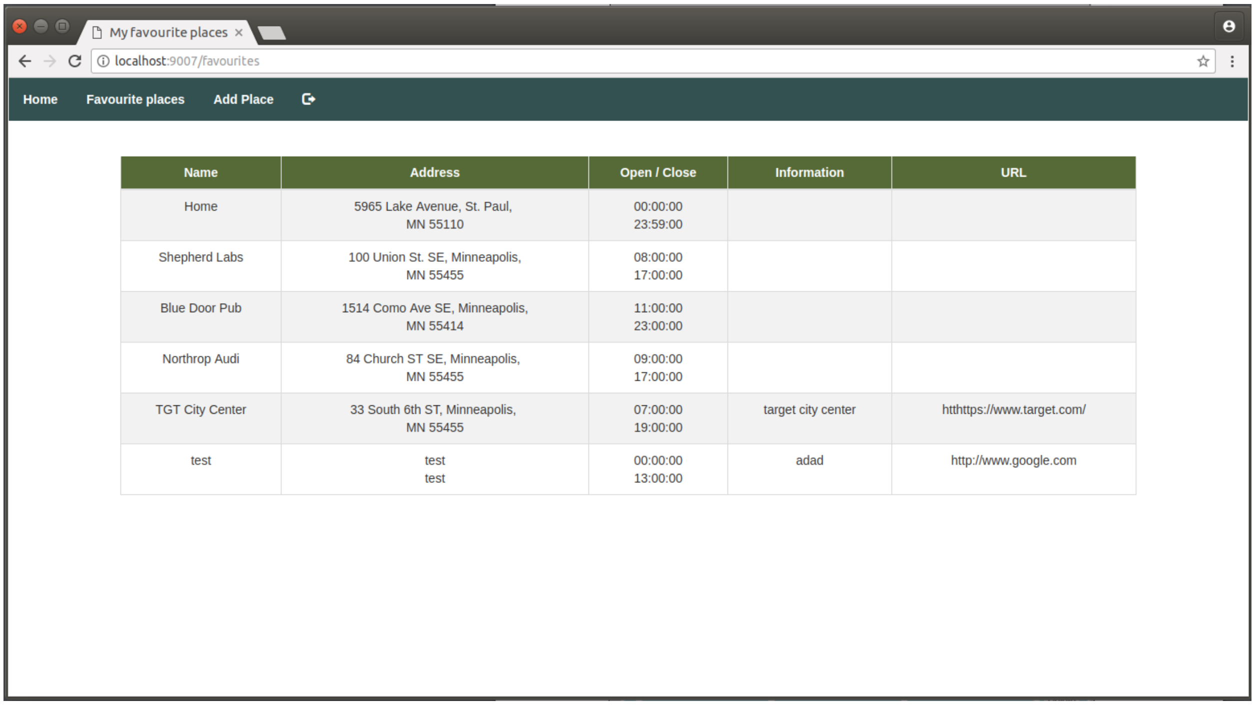
User:

invalid_user

Password:

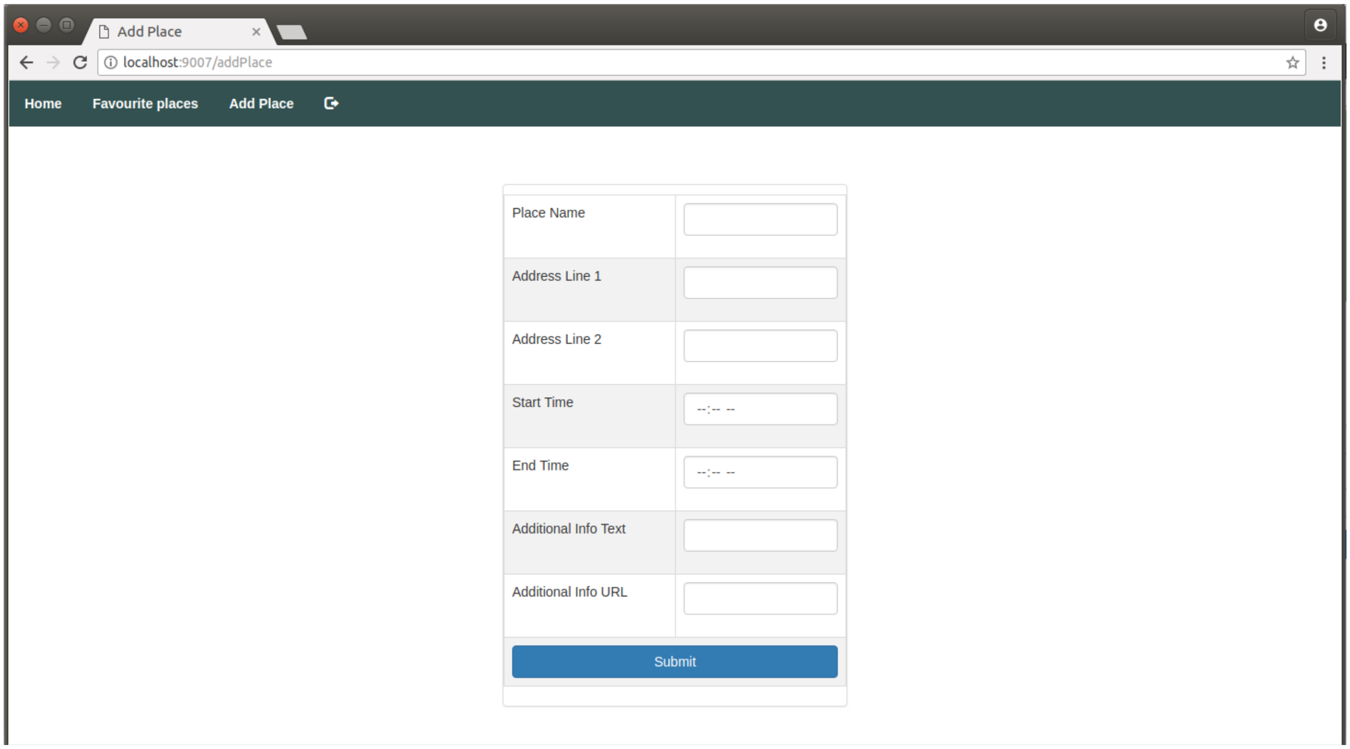
••••••••

Favorites Page

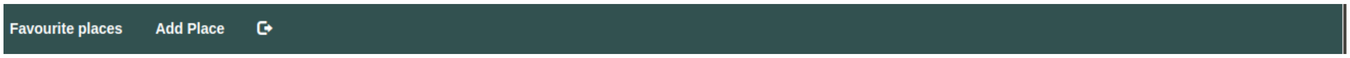


Name	Address	Open / Close	Information	URL
Home	5965 Lake Avenue, St. Paul, MN 55110	00:00:00 23:59:00		
Shepherd Labs	100 Union St. SE, Minneapolis, MN 55455	08:00:00 17:00:00		
Blue Door Pub	1514 Como Ave SE, Minneapolis, MN 55414	11:00:00 23:00:00		
Northrop Audi	84 Church ST SE, Minneapolis, MN 55455	09:00:00 17:00:00		
TGT City Center	33 South 6th ST, Minneapolis, MN 55455	07:00:00 19:00:00	target city center	htthttps://www.target.com/
test	test test	00:00:00 13:00:00	adad	http://www.google.com

Add Place page



Navigation bar


Favourite places
Add Place