CSCI 4131 – Internet Programming
Homework Assignment 6

Due Saturday November 27th at 6pm (NOTE: Change in Day / Time)
Late Submissions (with Penalty) accepted until Sunday, Nov. 28th 11:59pm

Description

Assignment 5 provided an introduction to web-development using Node.js. This assignment will build upon 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 server-side applications, and it is the most widely used Node.js application framework for doing so. Typical features of Express are:

- routing: a simple way to map URLs and http verbs to code paths on the Node.js server
- 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
- Additional References
  - Express website
  - Books and blogs
  - FAQ
  - Routing in Express
  - API Reference

This assignment will also introduce you to SQL and the MySQL database (so upon successful completion, you will have developed a FULL-STACK application).

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

- Your zyBook!!! Chapters 11 and 14
- https://www.w3schools.com/sql/
- https://www.w3schools.com/sql/sql_ref_mysql.asp
- https://www.w3schools.com/nodejs/nodejs_mysql.asp
- Optional SQL/MYSQL: Chapter 13 Sebesta

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 remotely (by SSH or VOLE).
2. Most of the CSE lab machines run version 12.18.3 (or similar version) of Node.js.
3. Type the following command to check the availability and the version of Node.js on the machine:

   ```
   node -v
   ```

   And this will display the current installed version.

4. To use the MYSQL database, you will need a database user id and password. Your MYSQL database user id (alphanumeric) and password (numeric) can be found on the class Canvas site in your Grades menu. Check the column named: **DATABASE INFORMATION**. You will find your ***alphanumeric*** user id (name) and ***numeric*** password in the comments section.

5. At the terminal, type the following command to login to MySQL and check whether it's active:

   ```
   mysql -uyour_database_user -hcse-mysql-classes-01.cse.umn.edu -P3306 -p your_database_user
   ```

   NOTE: There is a space between the lowercase –p and your_database_user id, and your database user id is used in 2 places

   Replace your_database_user with the database id provided to you before hitting enter. your_database_user will be in the format: C4131F21UXXX

   When prompted for a password, enter the NUMERIC password provided to you.

6. After successful login, you should see the prompt:
   mysql>

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 (in person, via Vole, or SSH)
2. Create a directory named <x500id_hw06> 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 guidelines on the next page 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. The listing of all the available files in the directory (obtained by typing `ls -al` followed by the `<enter>` key) should display the following entry:

```
-rw------- 1 yourid CS-Grad 209 Nov 11 17:33 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. The npm modules that will be useful for this assignment and should be installed are:

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

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

NOTE: We have provided a sample file for the server (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_events_table.js

2. Download the files listed above and move them to **`yourx500id_hw06`** directory.
3. Edit each of the files to include your database id and numeric password, which you can find on Canvas in your grades in the comments portion of column named: **DATABASE Information**
4. Set the permissions on the files and directories to `rwxr-xr-x` (i.e., **`chmod 755 filename`**)
5. 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.

6. 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 your choose (we have provided default values) for acc_login and acc_password to login to the website. Keep the values in a safe place and do not share them with anyone.

7. At the terminal, type the following command to create the MySQL table: tbl_events

   ```
   node create_events_table.js
   ```

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

You are now ready to start designing and implementing the website functionality for the assignment!!!!

3 Functionality

Your website will have 5 pages, and possibly 6:

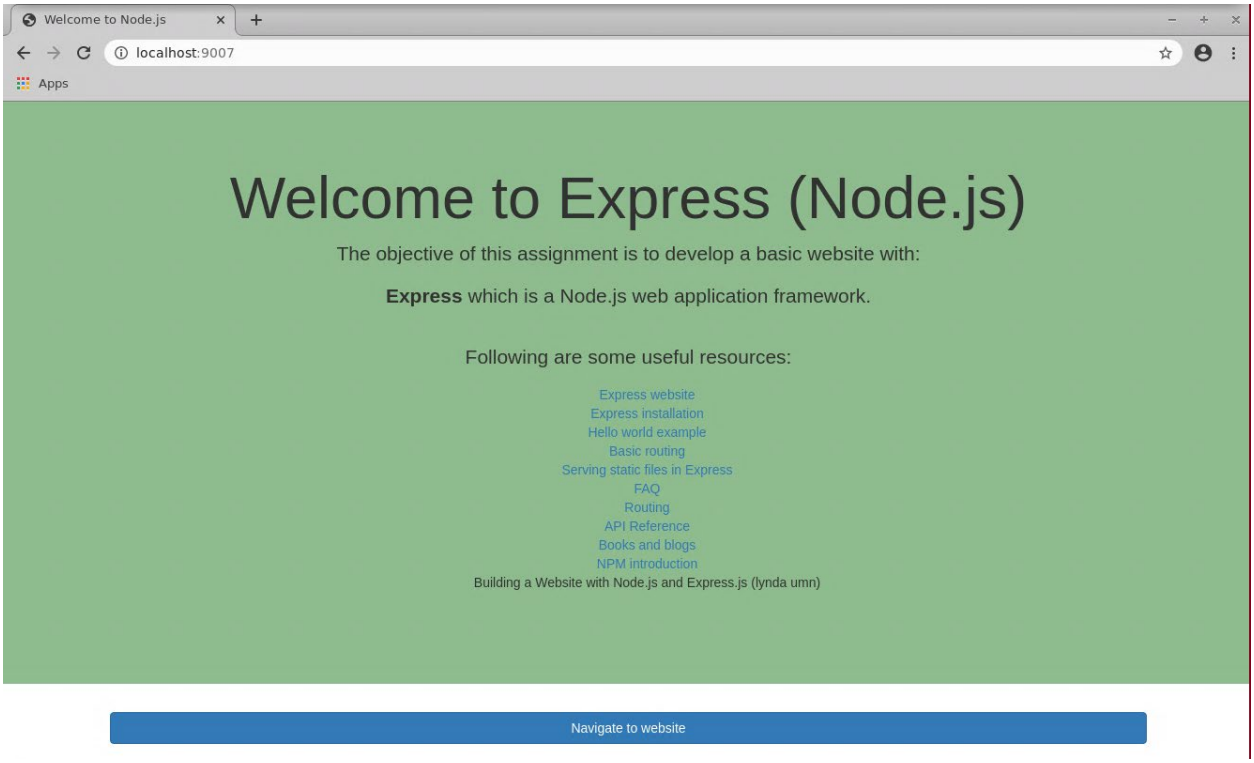
- A **Welcome** Page (provided)
- A **Login** page
- An **All Event** page
- A **Schedule** Page (You should be able to use your webpage from Homework 5)
- An **Add Event** page (You should be able to use (and for a bonus, update) your webpage from Homework 5).
- A **Stock** page (NOTE: IMPLEMENTING THIS FUNCTIONALITY IS A BONUS)

The All *Events*, Schedule, *Add Event*, *Stock* pages should have a navigation bar with a logout button.

NOTE: For this assignment you will need to develop the entire website including frontend (HTML pages, CSS, Javascript) and backend (Express server) + MySQL database.

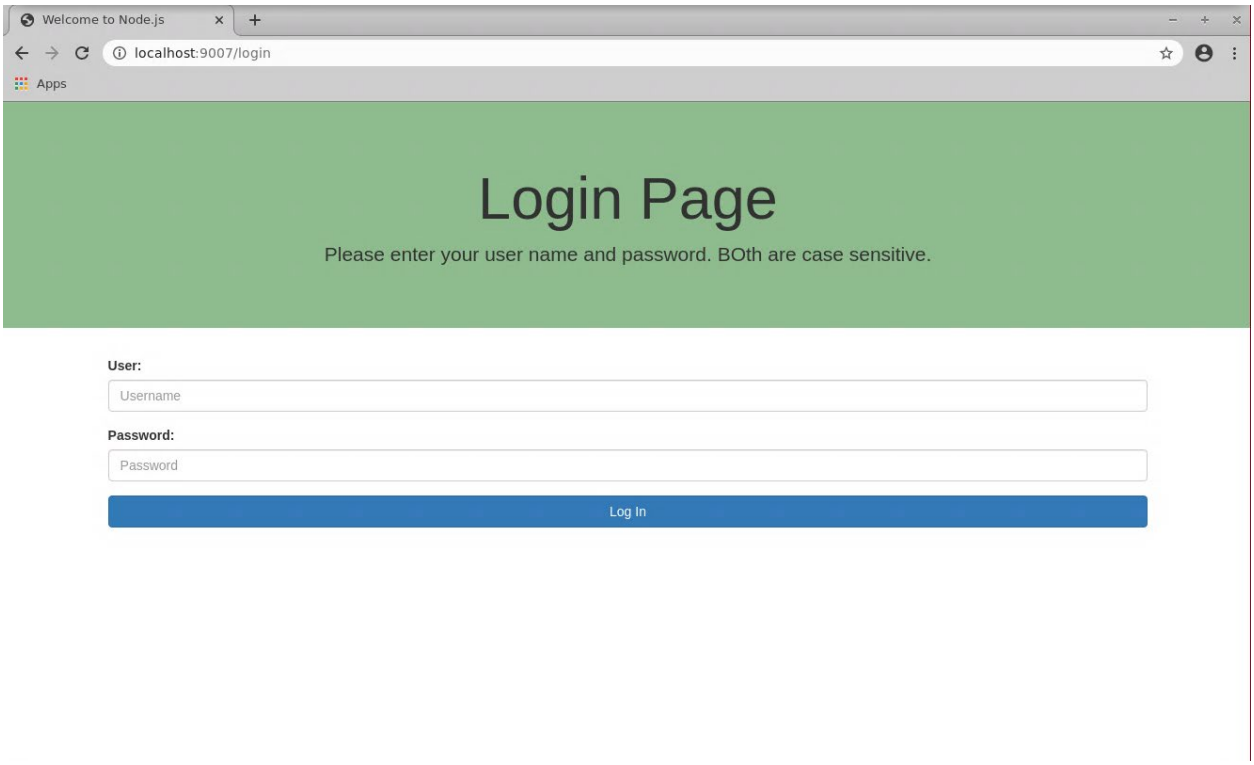
The pages below specify the functionality we have provided, and the functionality you must develop.

Welcome Page



- The Welcome page is already provided to you and is displayed when the default route "/" is called. (for example, by typing http://localhost:portnumber into your browser address bar)
- When you click on the Navigate to website button, the `/login` route in your Node.js/Express server (found in the file: `index.js` we have provided) will be called. You need to develop all the remaining functionality.

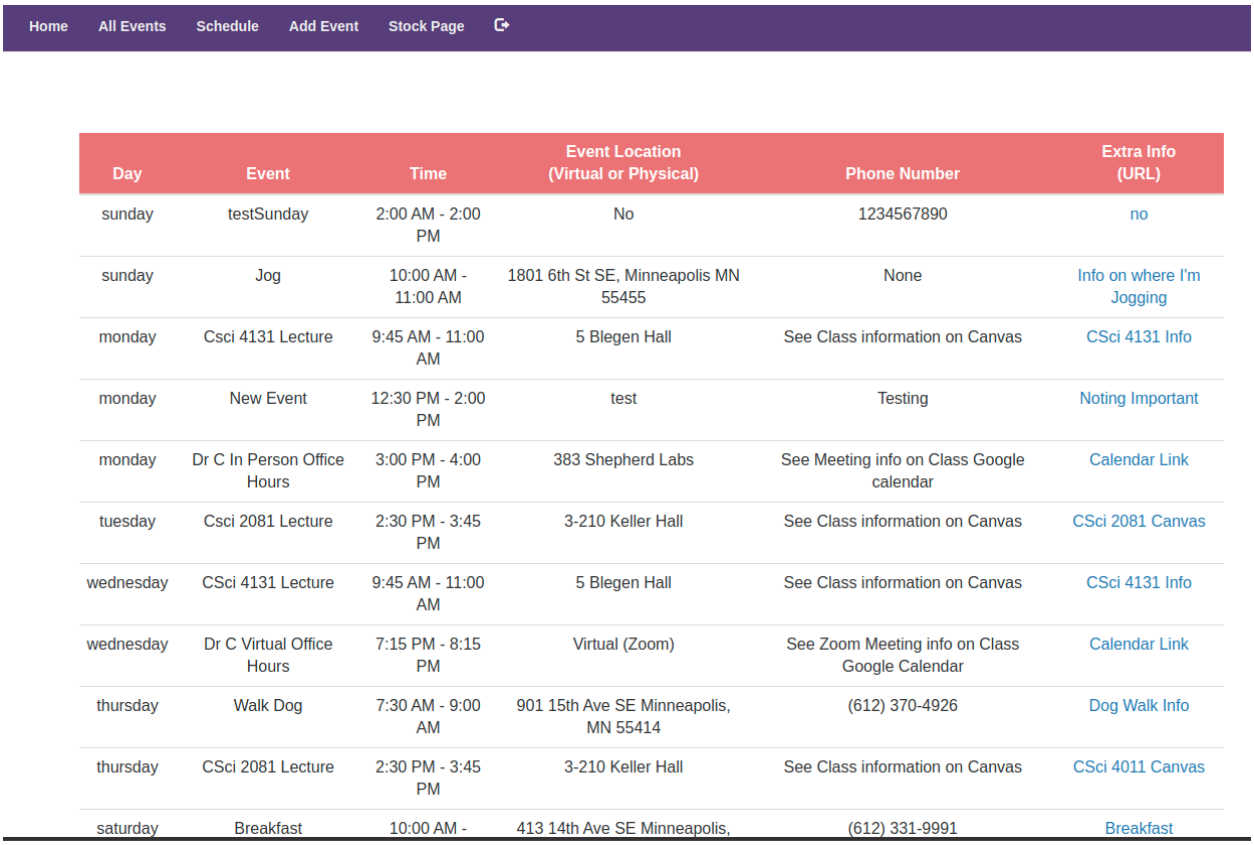
Login page



- If the page is accessed by a user that has already logged in, the user should be routed to the "All Events" page.
- The Login page should have a form with two fields: "User", and "Password"
- Both of these fields are mandatory (required).
- When the submit button is clicked, an AJAX request carrying the value entered for "User" and "Password" should be sent to the server for user credentials validation before allowing further access to the website.
- The server will validate the values obtained from the form against the *acc_login*, and *acc_password* fields stored in *tbl_accounts*. The Server should compare the bcrypt password-hash of the password string it obtains from the form with the one stored in the database table `tbl_accounts`.
    - ***Note:*** *your implementation muse use the bcrypt node module.*

- Upon successful validation, server should
    - Create a user session (*Note: this requires express-session module*).
    - Send a response back to the client indicating successful validation.
- If the validation fails, server should:
    - Send a response back to the client indicating validation failure.
- If successful response is received from server, user should be routed to "All Events" page, otherwise an appropriate error message should be displayed to the user (Check screenshots towards the end of this assignment)

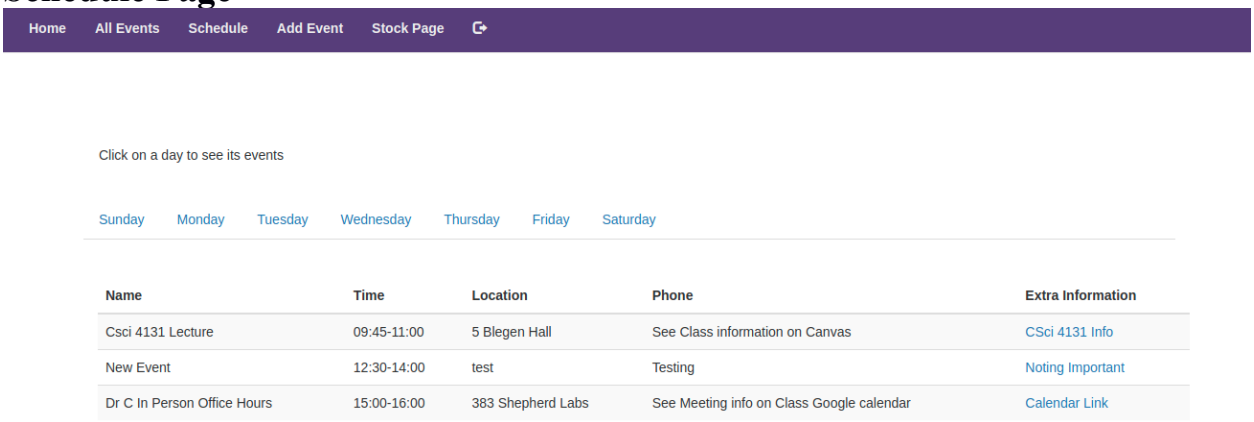
All Events page



Day	Event	Time	Event Location (Virtual or Physical)	Phone Number	Extra Info (URL)
sunday	testSunday	2:00 AM - 2:00 PM	No	1234567890	no
sunday	Jog	10:00 AM - 11:00 AM	1801 6th St SE, Minneapolis MN 55455	None	Info on where I'm Jogging
monday	Csci 4131 Lecture	9:45 AM - 11:00 AM	5 Blegen Hall	See Class information on Canvas	CSci 4131 Info
monday	New Event	12:30 PM - 2:00 PM	test	Testing	Noting Important
monday	Dr C In Person Office Hours	3:00 PM - 4:00 PM	383 Shepherd Labs	See Meeting info on Class Google calendar	Calendar Link
tuesday	Csci 2081 Lecture	2:30 PM - 3:45 PM	3-210 Keller Hall	See Class information on Canvas	CSci 2081 Canvas
wednesday	CSci 4131 Lecture	9:45 AM - 11:00 AM	5 Blegen Hall	See Class information on Canvas	CSci 4131 Info
wednesday	Dr C Virtual Office Hours	7:15 PM - 8:15 PM	Virtual (Zoom)	See Zoom Meeting info on Class Google Calendar	Calendar Link
thursday	Walk Dog	7:30 AM - 9:00 AM	901 15th Ave SE Minneapolis, MN 55414	(612) 370-4926	Dog Walk Info
thursday	CSci 2081 Lecture	2:30 PM - 3:45 PM	3-210 Keller Hall	See Class information on Canvas	CSci 4011 Canvas
saturday	Breakfast	10:00 AM -	413 14th Ave SE Minneapolis,	(612) 331-9991	Breakfast

- The "Home" button in the navigation bar will redirect the user to the Welcome page.
- If a user tries to access this page without a valid login, the user should be routed to the "**Login**" page.
- The page should have a navigation bar with a logout button.
- The table in this page should be dynamically populated.
- To achieve this, the server should provide a GET API (that is, a route) which returns the list of events. This API (route) will be very similar to the ones used by the menu in the client pages in assignment 5. It will get the list of events by querying the table: *tbl_events*.
- The client will call this route and populate the table using the data received from the server.

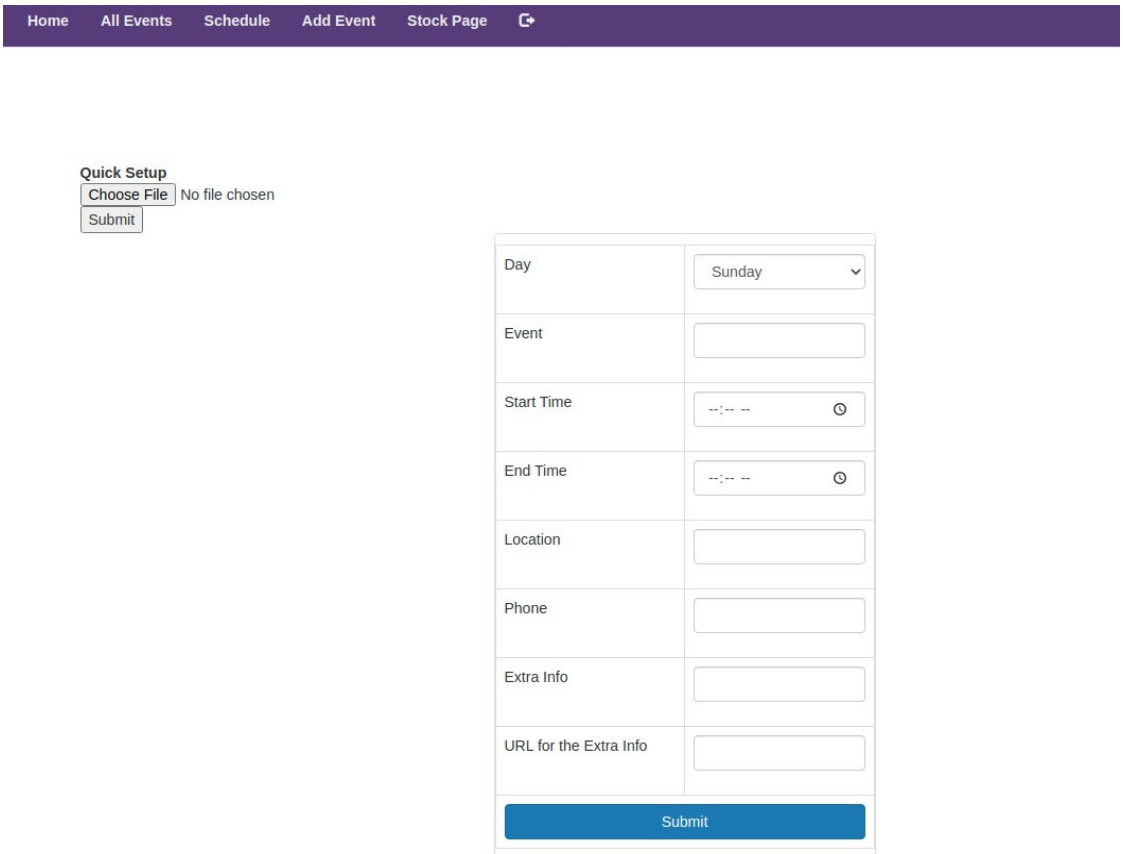
Schedule Page



This page should be very similar to the one developed in homework assignment 5. The screenshot displayed above shows the schedule after clicking “Monday”.

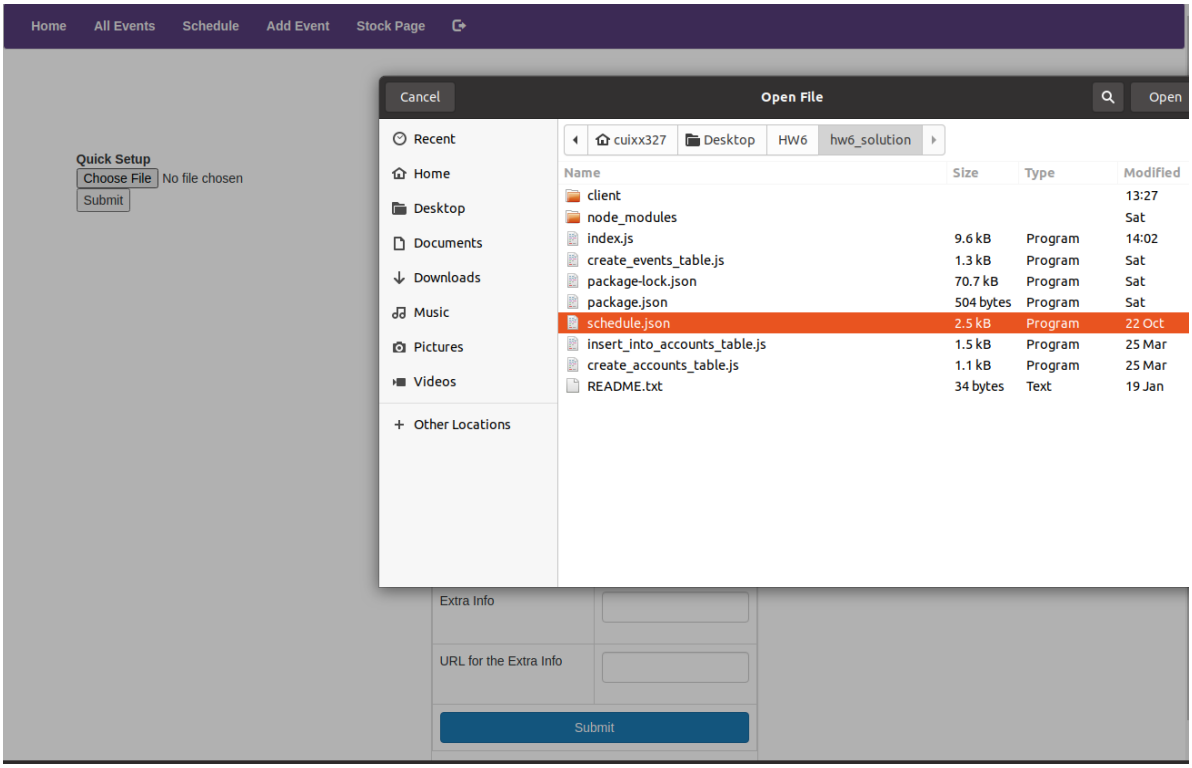
For this assignment, you will have to change the server functionality to retrieve the schedule information from the MySQL database and return it in a JSON object, instead of reading the schedule information from a local json file and returning it as was implemented in HW 5.
Note: you need to make sure the events in both All Events Page and Schedule Page are sorted in ascending order (by their day and start time, as shown on the screenshots).

Add Event page

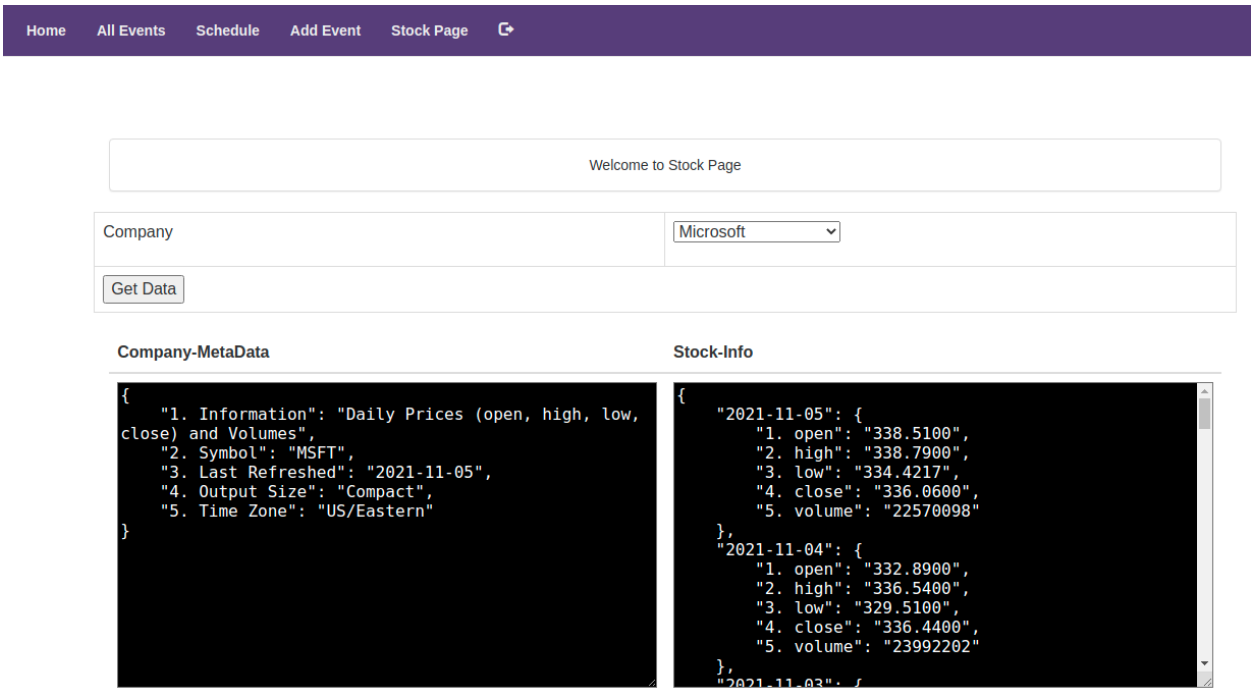


- You can use the form provided in Homework 5 for the 'Add Event' page.
- If this page is accessed without a valid login, the user should be routed to the "**Login**" page.
- The page should have a navigation bar with a logout button.
- Upon clicking submit, the form data should be posted to the server.
- The server should insert the received data into the following table: *tbl_events* instead of the file schedule.json used in HW 5 (*Hint: you will need use functionality provided by the mysql module*)
- The mapping between form fields and table columns could be (you are free modify this mapping as you deem necessary to comply with the functional requirements of this assignment):
    - day: event_day
    - event: event_event
    - start: event_start
    - end: event_end
    - location: event_location
    - phone: event_phone
    - info: event_info
    - url: event_url
- Upon successful insertion, the server should return a redirect response to the browser to display the "All Events" page.

- For BONUS points, add a “Quick Setup” (or whatever you want to call it) widget to facilitate easy and rapid importing of your schedule events from the file **schedule json** from HW5 to the database.
- Once the “Choose File” button is clicked, it should let you pick a file from your local file system, and after you select the **schedule.json** used by your HW5 assignment, clicking “Submit” will insert the events stored in the file you choose into the MySQL database (tbl_events) and then redirect to the “All Events” page.



BONUS Stock page (this screenshot was taken after “Get Data” being clicked)



- THIS PAGE IS AN OPTIONAL BONUS PAGE – your implementation should display stock quotes in a manner similar to the picture
- **You** must find a free API for getting stock quotes, and then use it to develop the functionality.
- If this page is accessed without a valid login, the user should be routed to the “**Login**” page.
- The page should have a navigation bar with a logout button.

Logout button

Upon clicking the logout button on the menu-bar (pictured below) , the session should be destroyed and the server should send a redirect message to the browser to display the **Login** page.

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**.

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. Finally, compress (e.g., tar or zip) the **yourx500id_express** directory and submit it **via Canvas.**

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.

Please remove the node_modules/ folder from your submission! Do not leave it in your submission!

AND, submit something no later than the late submission deadline to ensure you get credit any functionality you have working. Submissions after the late submission deadline will not be accepted, and will be assigned a grade of ZERO.

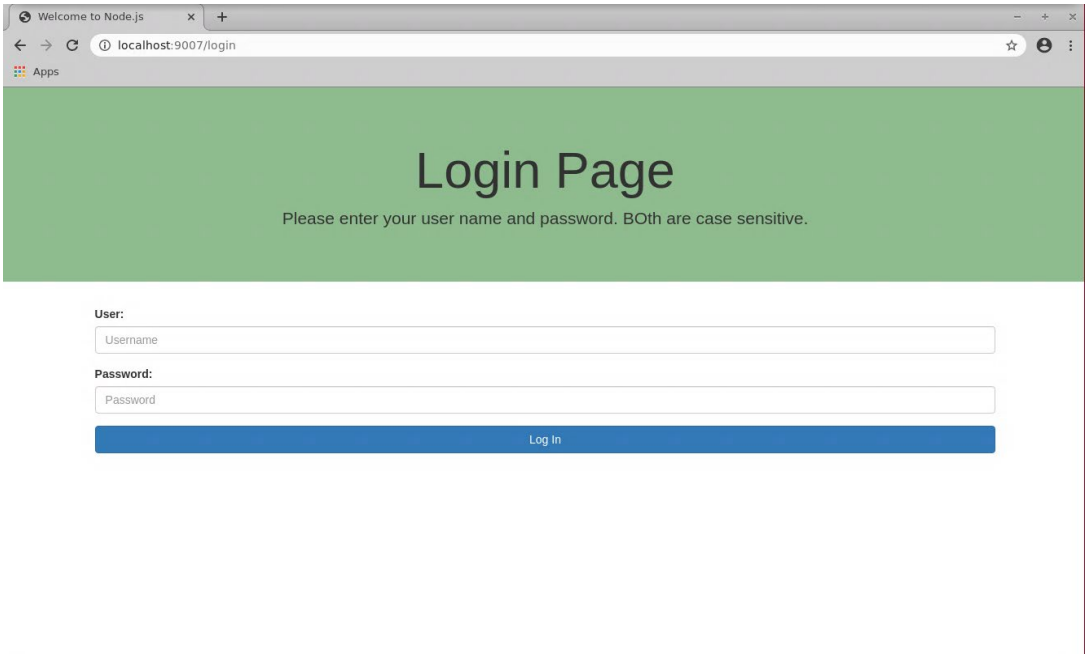
Evaluation

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

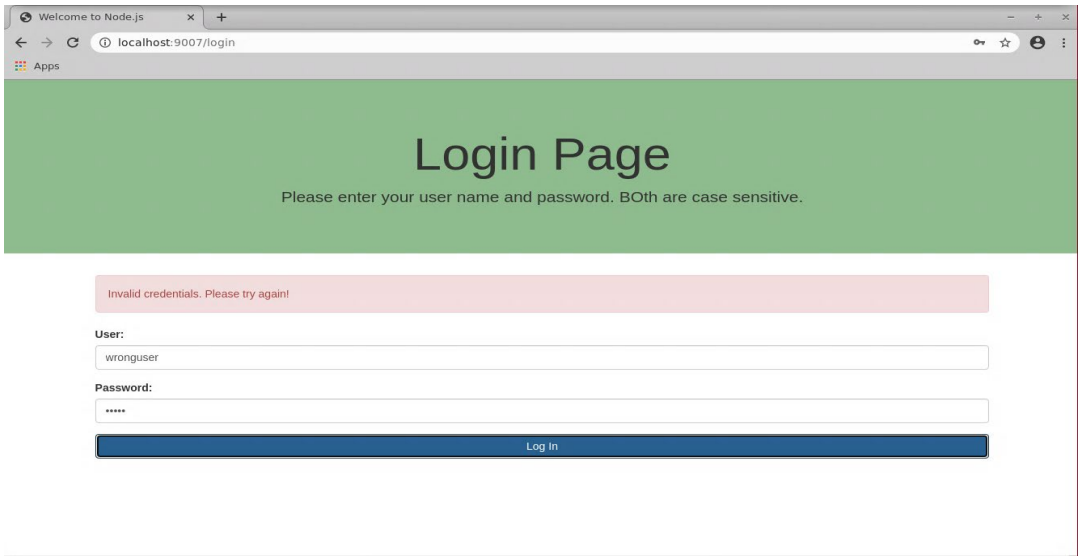
1. Submission instructions are met. **SPECIFICALLY** - remember to remove the **node_modules** folder **(5 points, really 5 points!)**
2. The "All Events" and “Schedule” and "Add Event" and “Stock” pages of your website redirect the user to "Login" page automatically before authentication. **(10 points)**
3. The "Login" page shows the form elements and the submit button. **(5 points)**
4. Use AJAX login to send user credentials to the server, and redirect the user to the “Events” page after successful login validation by the server. (check the additional screenshots) **(10 points)**
5. If server login validation fails, an error message is displayed on the "Login" page and the browser displays the login page and error message. **(10 points)**
6. After successful login, the "All Events" page displays correctly. **(5 points)**
7. The "All Events" page gets the list of events from the server (which the server gets from the database). These events are dynamically added to the table displayed in the user’s browser. **(10 points)**
8. The "Schedule" page gets the list of events from the server (which the server gets from the database). These events are dynamically added to the table displayed in the user’s browser. **(10 points)**
9. The events shown on “All Events” and “Schedule” pages are correctly sorted. **(10 points)**

10. The user can add a new event to the database using the form present in the "Add Event" page. **(10 points)**
11. The “All Events” and “Schedule” and “Add Event” and, if implemented, “Stock” pages display and have an operational navigation bar. **(5 points)**
12. When a new event is added through the “Add Event” page, the event data is stored in the MySQL database. Then the user is redirected to the "Events" page, and the user’s events are correctly displayed. **(5 points)**
13. The logout functionality works correctly. **(5 points)**
14. **The BONUS stock page works correctly. (15 point BONUS)**
15. **The BONUS Quick Setup functionality works correctly. (10 point BONUS)**

Additional Screenshots (See the following pages for examples)



Invalid Login


Welcome to Node.js
localhost:9007/login
Apps
Login Page
Please enter your user name and password. BOth are case sensitive.
Invalid credentials. Please try again!
User:
wronguser
Password:
•••••
Log In