

# CSCI 4131 – Internet Programming

## Homework Assignment 6 (Version 1, November 14)

**Due Monday, Nov 30<sup>th</sup> at 11:59pm**

**Late Submissions accepted with Penalty after Due Date through Tuesday at 11:59pm**

**This an individual assignment. Do your own work**

### Description

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

- Simplified routing: a simplified way to map URLs and http verbs to server side code paths
- Simplified methods for parsing http requests and building http responses
- Simplified support for sessions

**Note, Homework 7 will build on this assignment, and the final project will require a version of HW 7 with the functionality in this assignment (HW 6) working, at a minimum.**

*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 Referenced
  - [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

- SQL/MYSQL: Chapter 13 Sebesta
- <https://www.w3schools.com/sql/>
- [https://www.w3schools.com/sql/sql\\_ref\\_mysql.asp](https://www.w3schools.com/sql/sql_ref_mysql.asp)
- [https://www.w3schools.com/nodejs/nodejs\\_mysql.asp](https://www.w3schools.com/nodejs/nodejs_mysql.asp)

## 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 and numeric password can be found on the class Canvas site. The user name and password are separated by a space.
5. At the terminal, type the following command to login to MySQL and check whether it's active:

```
mysql -uyour_database_login -hcse-larry.cse.umn.edu -P3306 -p your_database_login
```

You will find your database login and NUMERIC database password on Canvas with your grades in the feedback section of the column named: **Mysql database login and password**

Replace your\_database\_login with the database login provided to you before hitting enter. your\_database\_login will be in the format: **C4131F20UXXX**

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

6. After successful login, you should see **mysql>** prompt. Logout with the **exit** command.

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\_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 specified on the next page to enter the information (The information you need to enter is specified 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:

```
-rw----- 1 he000242 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 these files 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: **Mysql database login and password**.
4. Set the permissions on the files to rwxr-xr-x (e.g., **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 chosen 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.

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

### 3 Functionality

Your website will have 5 pages:

- A **Welcome** Page (provided)
- A **Login** page
- A **Event** page
- An **Add Event** page
- A **Stock** page

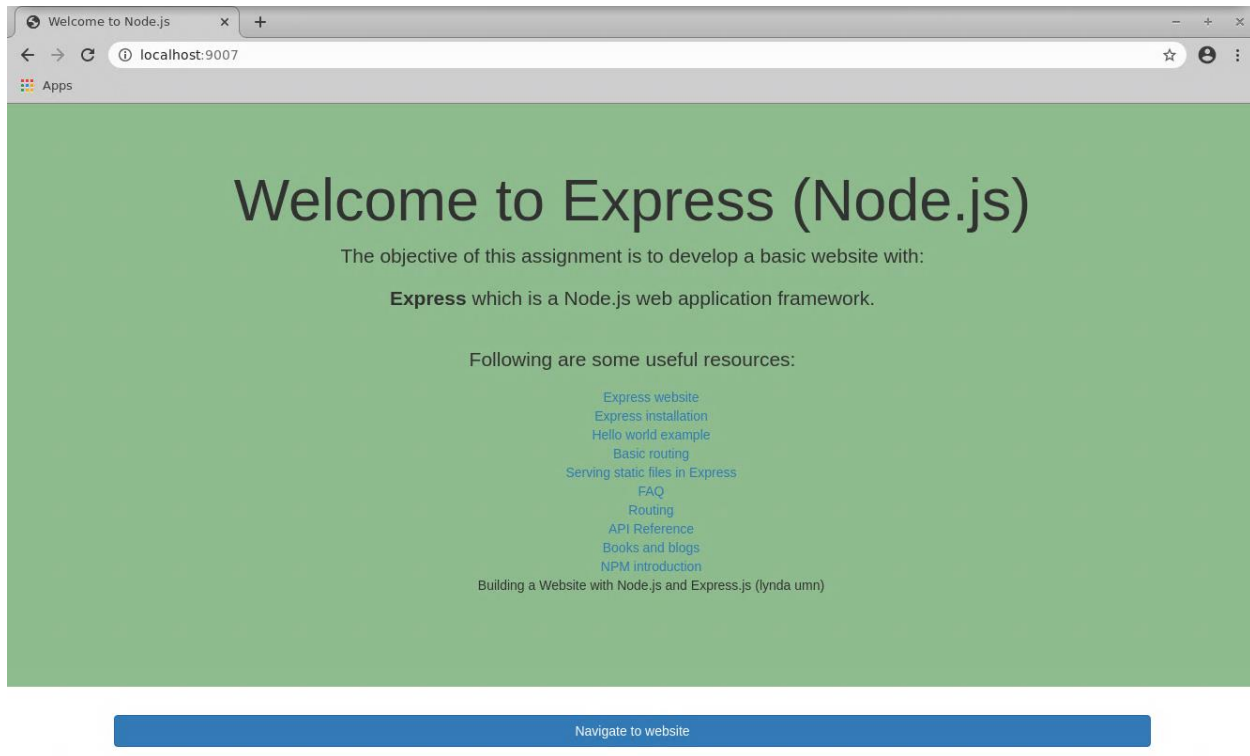
The *Event*, *Add Event*, *Stock* pages will have a **navigation bar with a logout button**.

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

**Pages 5 through 9 below specify the functionality provided, and the functionality you must develop.**

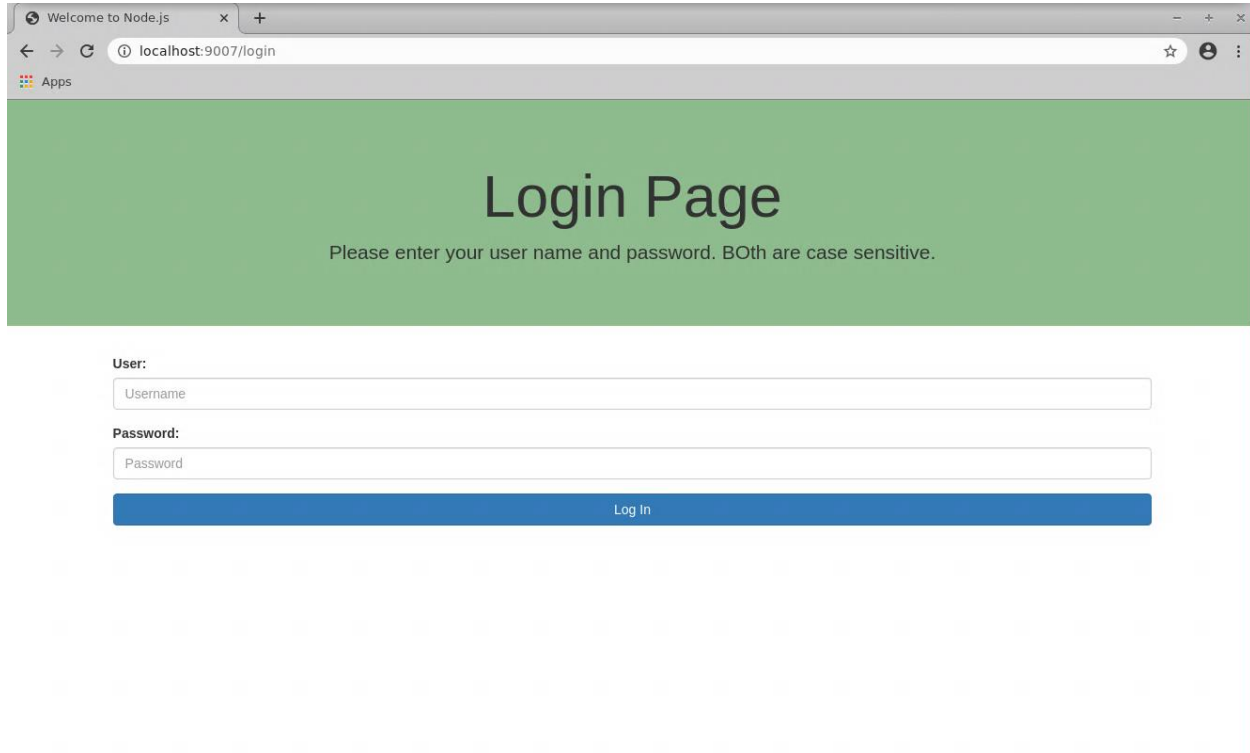
**NOTE: you should be able to use the Event page, Add Event Page, and Stock page from Homework 5 – but you will have to add the logout menu item and functionality**

## Welcome Page (Provided)



- The Welcome page is already provided to you and is displayed when the default route “/” is called.
- When you click on the Navigate to website button, the /login route on the Express (Node.js) will be called. You need to develop all the remaining functionality.

## Login page (You need to create this page and all functionality for it)



Welcome to Node.js x +

localhost:9007/login

# Login Page

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

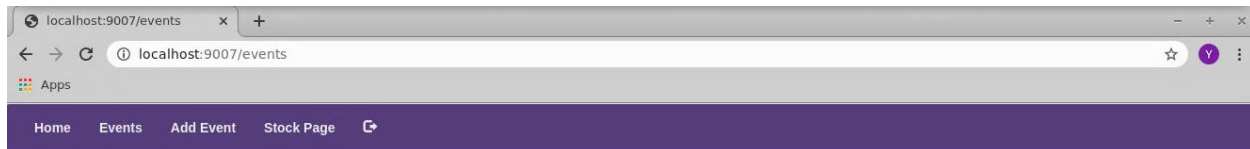
User:

Password:

Log In

- If the page is accessed by a user that has already logged in, the user should be routed to the “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. Passwords are stored in the MySQL database in a SHA256 hashed format in tbl\_accounts. The Server should hash the password string it obtains from the form using the SHA256 hash algorithm and compare it to the SHA256 hashed password stored in the database table tbl\_accounts.  
*(Hint: you can use crypto module to create the SHA256 hash. An example indicating how to use the SHA256 hashing algorithm can be found in the file we gave you to insert login information into the database named: **insert\_into\_accounts\_table.js**)*
- Upon successful validation, server should
  - Create a user session (*Hint: you can use 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 “Events” page, otherwise an appropriate error message should be displayed to the user (Check screenshots towards the end of this assignment)

## Events page (You should be able to reuse the page from HW 5 and update it with the logout menu item)



Day	Event	Time	Event Location (Virtual or Physical)	Phone Number	Extra Info (URL)
Monday	Csci 4131 Lecture	9:45 AM - 11:00 AM	Virtual (Zoom)	See Zoom Meeting info on Class	<a href="#">Ccsi 4131 Info</a>
Monday	Csci 2081 Lecture	1:25 PM - 2:15 PM	Virtual (Zoom)	See Zoom Meeting info on Class	<a href="#">Ccsi 2081 Info</a>
Tuesday	Virtual Office Hours	12:00 PM - 1:00 PM	Virtual (Zoom)	See Zoom Meeting info on Class	<a href="#">Ccsi 2081 Info</a>
Wednesday	Csci 4131 Lecture	9:45 AM - 11:00 AM	Virtual (Zoom)	See Zoom Meeting info on Class	<a href="#">Ccsi 4131 Info</a>
Wednesday	Csci 2081 Lecture	1:25 PM - 2:15 PM	Virtual (Zoom)	See Zoom Meeting info on Class	<a href="#">Ccsi 2081 Info</a>
Thursday	Walk Dog	7:30 AM - 9:00 AM	901 15th Ave SE Minneapolis, MN 55414	(612) 370-4926	<a href="#">Dog Walk Info</a>
Friday	Csci 2081 Lecture	1:25 PM - 2:15 PM	Virtual (Zoom)	See Zoom Meeting info on Class	<a href="#">Ccsi 2081 Info</a>
Saturday	Breakfast	10:00 AM - 11:00 AM	413 14th Ave SE Minneapolis, MN 55414	(612) 331-9991	<a href="#">Breakfast Spot</a>
Sunday	Jog	10:00 AM - 11:00 AM	1801 6th St SE, Minneapolis MN 55455	None	<a href="#">Info on where I'm Jogging</a>

- 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 which returns the list of events. This API will be very similar to the one developed in assignment 5. It will get the list of events by querying the table: *tbl\_events*.
- The client will call this API and populate the table using the data received from the server.

## Add Event page (You should be able to reuse your page from HW 5 and update it with the logout menu item)

The screenshot shows a web browser window with the address bar displaying 'localhost:9007/addEvent'. The browser's address bar also shows a search icon, a star icon, and a user profile icon. Below the address bar is a purple navigation bar with the following links: Home, Events, Add Event, and Stock Page. The main content area of the page is white and contains a form with the following fields:

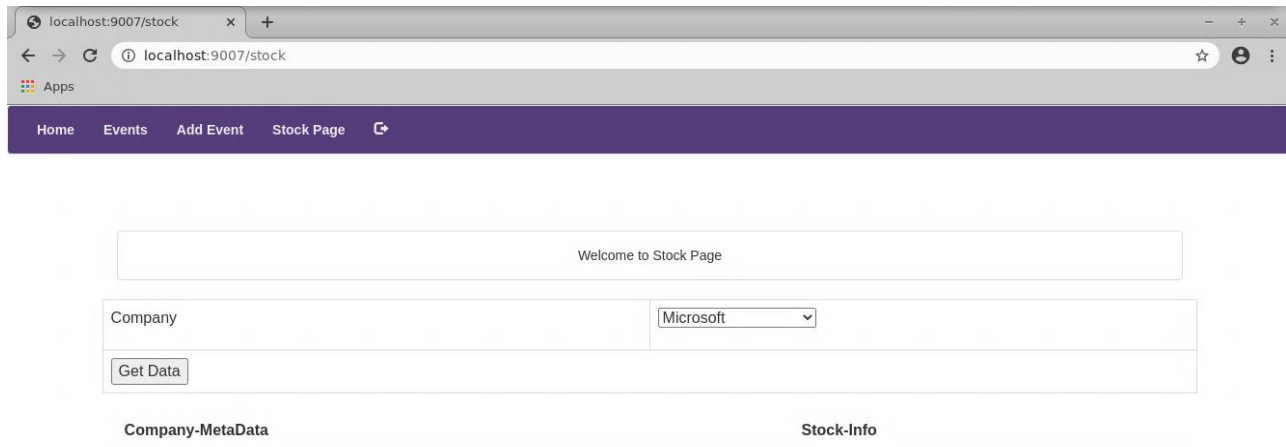
- Day: A dropdown menu with 'Sunday' selected.
- Event: A text input field.
- Start Time: A time selection field.
- End Time: A time selection field.
- Location: A text input field.
- Phone: A text input field.
- Extra Info: A text input field.
- URL for the Extra Info: A text input field.

At the bottom of the form is a blue 'Submit' button.

- You can use the form developed in the earlier assignments 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 in your MySQL database: *tbl\_events* (*Hint: you can use mysql module*)
- The mapping between form fields and table columns is:
  - 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 “Events” page.



## Stock page (You should be able to reuse your page from HW 5 and update it with the logout menu item)



localhost:9007/stock

Home Events Add Event Stock Page

Welcome to Stock Page

Company Microsoft

Get Data

Company-MetaData Stock-Info

- You can use the ‘Stock’ page developed in the earlier assignments.
- 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 (pictured below) on the menu-bar of the Events, Add Event, and Stock pages, 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!**

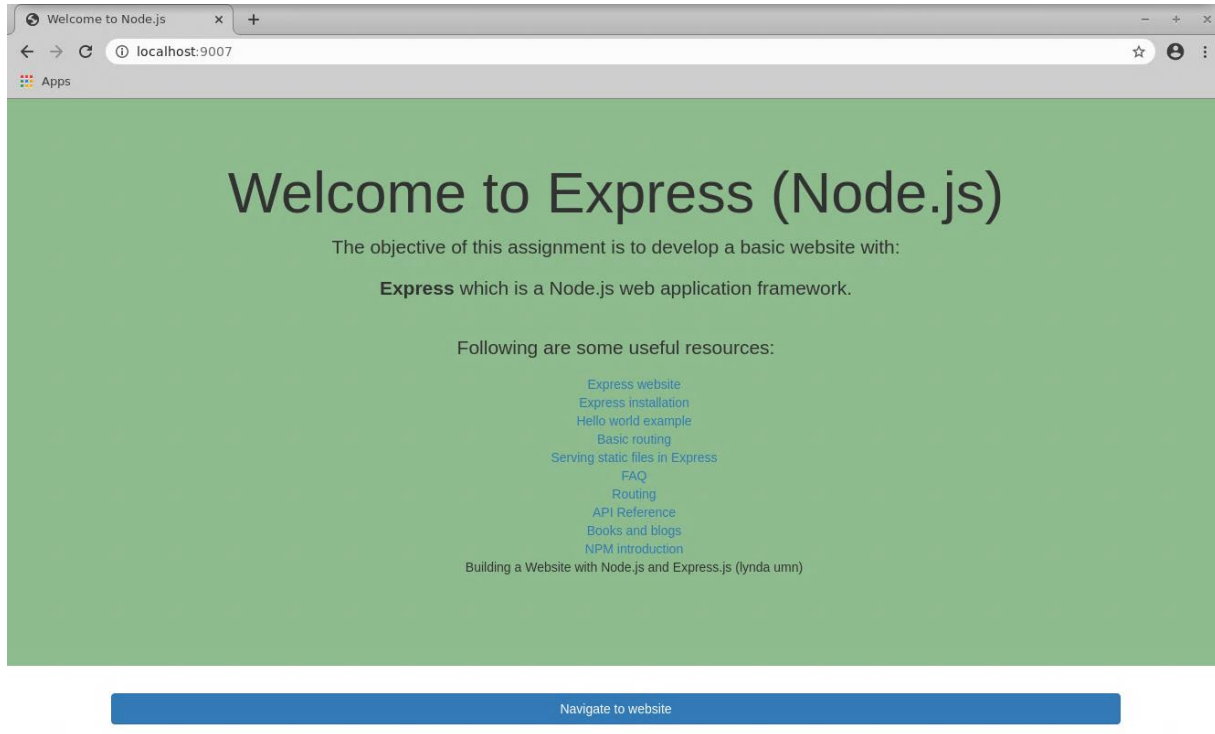
## Evaluation

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

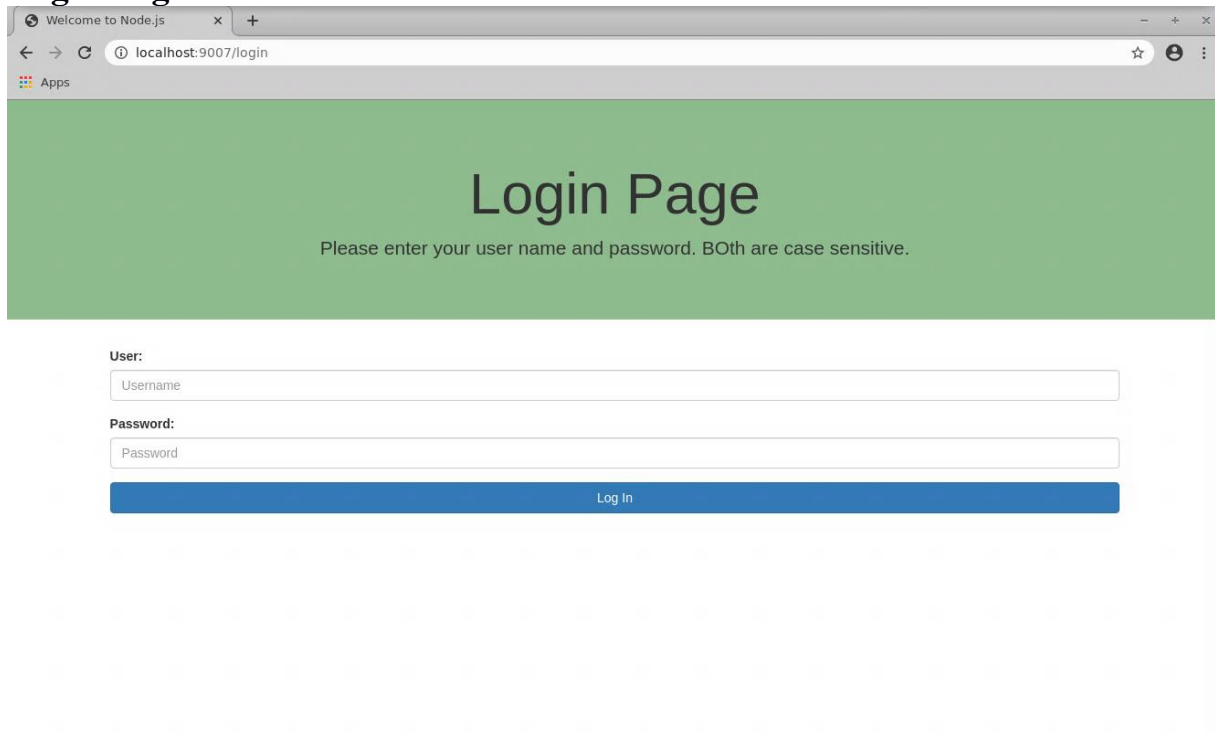
- **Submission instructions are met. (5 points)**
- The "Events" and "Add Event" and "Stock" pages of your website redirect the user to "Login" page automatically before authentication. **(10 points)**
- The "Login" page shows the form elements and submit button. **(5 points)**
- Use AJAX login to send user credentials to the server, and redirect the user to the "Events" page after successful login validation by the server. **(10 points)**
- 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)**
- After successful login, the "Events" page displays correctly and has an operational navigation bar. **(5 points)**
- The "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. **(20 points)**
- The user can add a new event to the database using the form present in the "Add Event" page. **(15 points)**
- The "Add Event" page displays and has an operational navigation bar. **(5 points)**
- 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)**
- The "Stock" page displays and has an operational navigation bar. **(5 points)**
- The logout functionality works correctly. **(5 points)**

## Additional Screenshots (See the following pages for examples)

### Welcome Page



### Login Page



## Invalid Login

The screenshot shows a web browser window with the address bar at `localhost:9007/login`. The page has a green header with the text "Login Page" and "Please enter your user name and password. BOTH are case sensitive." Below this, there is a red error message box that says "Invalid credentials. Please try again!". Underneath the error message, there are two input fields: "User:" with the value "wronguser" and "Password:" with the value "\*\*\*\*\*". At the bottom, there is a blue "Log In" button.

Welcome to Node.js x +

localhost:9007/login

Apps

# Login Page

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

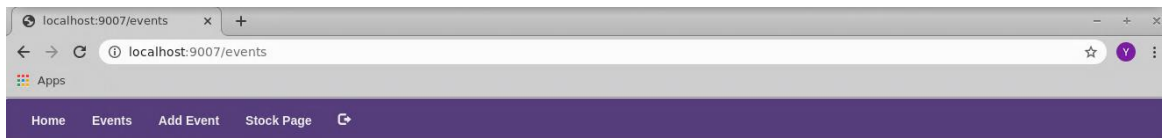
Invalid credentials. Please try again!

User:

Password:

Log In

## Events Page



Day	Event	Time	Event Location (Virtual or Physical)	Phone Number	Extra Info (URL)
Monday	Csci 4131 Lecture	9:45 AM - 11:00 AM	Virtual (Zoom)	See Zoom Meeting info on Class	<a href="#">Ccsi 4131 Info</a>
Monday	Csci 2081 Lecture	1:25 PM - 2:15 PM	Virtual (Zoom)	See Zoom Meeting info on Class	<a href="#">Ccsi 2081 Info</a>
Tuesday	Virtual Office Hours	12:00 PM - 1:00 PM	Virtual (Zoom)	See Zoom Meeting info on Class	<a href="#">Ccsi 2081 Info</a>
Wednesday	Csci 4131 Lecture	9:45 AM - 11:00 AM	Virtual (Zoom)	See Zoom Meeting info on Class	<a href="#">Ccsi 4131 Info</a>
Wednesday	Csci 2081 Lecture	1:25 PM - 2:15 PM	Virtual (Zoom)	See Zoom Meeting info on Class	<a href="#">Ccsi 2081 Info</a>
Thursday	Walk Dog	7:30 AM - 9:00 AM	901 15th Ave SE Minneapolis, MN 55414	(612) 370-4926	<a href="#">Dog Walk Info</a>
Friday	Csci 2081 Lecture	1:25 PM - 2:15 PM	Virtual (Zoom)	See Zoom Meeting info on Class	<a href="#">Ccsi 2081 Info</a>
Saturday	Breakfast	10:00 AM - 11:00 AM	413 14th Ave SE Minneapolis, MN 55414	(612) 331-9991	<a href="#">Breakfast Spot</a>
Sunday	Jog	10:00 AM - 11:00 AM	1801 6th St SE, Minneapolis MN 55455	None	<a href="#">Info on where I'm Jogging</a>

## Add Event page

The screenshot shows a web browser window with the address bar displaying 'localhost:9007/addEvent'. The browser's address bar also shows 'localhost:9007/addEvent' and a search icon. Below the address bar is a navigation bar with links: Home, Events, Add Event, and Stock Page. The main content area contains a form with the following fields:

Day	Sunday
Event	
Start Time	
End Time	
Location	
Phone	
Extra Info	
URL for the Extra Info	

Below the form is a blue Submit button.

## Stock page

The screenshot shows a web browser window with the address bar displaying 'localhost:9007/stock'. The browser's address bar also shows 'localhost:9007/stock' and a search icon. Below the address bar is a navigation bar with links: Home, Events, Add Event, and Stock Page. The main content area contains a form with the following fields:

Welcome to Stock Page	
Company	Microsoft
Get Data	

Below the form are two sections: Company-MetaData and Stock-Info.

## Navigation bar

The screenshot shows a navigation bar with the following links: Home, Events, Add Event, and Stock Page. There is also a search icon.