ITC5315: Project Status Report WK 2

Created by: Meenu Mathew

Student ID: N01582144

Date: 15-04-2024

# Table of contents

[Table of contents 1](#_Toc1122846617)

[Step 3 2](#_Toc557882305)

[Step 4 3](#_Toc499316495)

[Step 5 4](#_Toc1509694456)

[Step 6 5](#_Toc1563392119)

[Summary 6](#_Toc1613282810)

## Step 3

Describe your process for building the UI and Form support using a template engine. Describe which fields you provided in the form. Describe the CSS/styling that you applied. Describe any challenges you faced and how you approached solving them. Show screenshots of your UI. Show screenshots of testing the new route using Postman

I added page number, perPage and Borough values in the form. When I submit the form it will fetch corresponding page with restaurant details.

To start the UI I have to define a project structure with public folder and view folder. A template is also defined in it. I selected the handlebars template engine as the view engine. Then created routes and created view pages like index.hbs, search\_restaurants.hbs. Added a style.css file to design the page.

A background image added to the web page to make it more appealing.

A screenshot of a search bar

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

With Postman

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Step 4

Describe your approach to providing security in your application – Did you make any changes to the database? Describe the changes you made to support authorization of users. Show screenshots of testing authorization using Postman or your Browser.

First, I installed the necessary npm packages jsonwebtoken and bcryptjs for handling JWT and password hashing, respectively. Then I added a user’s table to database, and defined user model and schema, so users can register with username and password. Then they can login to the system. Only authenticated users will be allowed to view restaurant data. Otherwise, they will be redirected to login page.

For user registration, I created a POST route that accepts name, username and password, creates a new user object, and saves it to the database after hashing the password. For user login, I implemented a POST route that finds the user by username, compares the hashed password using bcrypt, and if valid, generates a JWT token using jsonwebtoken with a secret key and saved it in cookie. As I am implementing both front end and back end in a single system, I am using cookies for authentication.

To protect routes that require authentication, I implemented a middleware function called verifyToken that extracts the JWT from cookies, verifies it using the secret key, and decodes it to retrieve the user ID. If the token is valid, the middleware attaches the user ID to the request object, allowing access to the protected route.

I stored the JWT secret key and other sensitive information like database credentials in environment variables for added security. Additionally, I implemented error handling to manage JWT verification errors and invalid tokens, and if user fails to authenticate it will redirect to login page.

By following this approach, I enhanced the security of the Node.js application by implementing JWT-based authentication and authorization, password hashing, and other best practices to protect against threats.

**User Table**

A screenshot of a computer program

Description automatically generated

**Index/Login Page**

A screenshot of a computer

Description automatically generated

**Signup page**

A screenshot of a computer

Description automatically generated

**Signup validation for password**

**A screenshot of a computer

Description automatically generated**

**After successful signup user will be redirected to login page**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Then it will show the restaurant related pages.**

**In postman**

**A screenshot of a computer

Description automatically generated**

**As there is no authentication provided it will show error.**

**A screenshot of a computer

Description automatically generated**

**After login submission and authentication, it will load data.**

**A screenshot of a computer

Description automatically generated**

## Step 5

Describe the new feature you added and why to choose that feature. Show screenshots of you demonstrating the new feature in your browser. Describe any challenges you faced with the new feature you added.

A screenshot of a computer

Description automatically generated

As a new feature I gave a view link so the users can click on it and view all details about the restaurant.

A screenshot of a website

Description automatically generated

A back button is provided in most of the pages, so the users can redirect back and forth.

**Challenge**

The redirection was not working perfectly as the url was not loading properly. So defined a new parameter for base url in app.js.

A close-up of a computer code

Description automatically generated

A BASE\_URL parameter is defined in app.js and it is called in view pages to give links to other pages.



## Step 6

Describe any challenges you faced deploying to Vercel. Show screenshots of your application running on a Vercel domain. Provide the public URL of your application on Vercel (make sure it does not require authentication)

## Summary

Summarize your experience while working on this project for the second week. Would you have done anything differently? Did anything take shorter/longer than you expected?