Implementation Document

for

KnowYourCibo

Version 1.0

Prepared by

Group #: 6 **Group Name: Code Crafters**

K S U Rithwin	220537	ksupendra22@iitk.ac.in
R V S Havish	220879	rviswas22@iitk.ac.in
Mahanthi Vijay Kumar	220602	mvijay22@iitk.ac.in
G N L Mahathi	220395	gnagal22@iitk.ac.in
Rohan Aditya	220741	praditya22@iitk.ac.in
Rushikesh Chary	220336	drushikesh22@iitk.ac.in
Raj Vinayak Meena	220861	rajvm22@iitk.ac.in
Laveti Bhanu Prakash	220583	lbhanu22@iitk.ac.in
Sai Dishanth	220282	banoths22@iitk.ac.in
Ravula Harshith Sai	220878	ravula22@iitk.ac.in
Gowtham Chand	220313	cgchand22@iitk.ac.in

Course: CS253

Mentor TA: *Sarthak*

Date: 15 March, 2024

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.0	Group-Code Crafters	Initiated the document. Added all the necessary details.	18/03/2024

1 Implementation Details

Frontend

- We have used **HTML** (HyperText Markup Language) as a standard markup language for web pages to be displayed in the web browser. It provides a basic structure for web content, including text, images, and other media.
- We have used **CSS** (Cascading Style Sheets) to style web pages.
  - ➢ It allows for the separation of content and presentation, making it easier to maintain and update the web pages.
- We have used **JavaScript**
  - ➢ It allows us to create dynamic and interactive user interfaces on the client side.
  - ➢ It ensures website responsiveness.

DataBase

- We found **MongoDB** to be more useful than other database systems as we had so much unstructured data..
- It supports a powerful query language that includes a wide range of operators. This makes it easy to retrieve and manipulate data in a way that meets the specific requirements of applications.
- Its automatic sharding simplifies the process of distributing data across multiple servers to ensure high availability and fault tolerance.

Backend

- **Node.js** source code is freely available under an open-source license. This allows us the freedom to use, modify, and implement it as per our business needs.
- **Node.js** allows us to use JavaScript for both frontend and backend development.
- It's highly scalable. It can handle a large number of concurrent connections and scale horizontally by adding more servers.
- Asynchronous programming is a key feature of **Node.js** which makes it well-suited for building scalable and high-performance applications.
- It supports rapid development with features like hot module reloading, allowing for quick iteration and efficient code updates during the development process.

Here's how we've utilized **Node.js** and its ecosystem in our backend:

- Express.js: Leveraging this minimal and flexible Node.js web application framework, we have constructed a robust API that serves our frontend application efficiently.
- Mongoose: As an Object Data Modeling (ODM) library for MongoDB and Node.js, Mongoose provides a straightforward schema-based solution to model our application data. It includes built-in type casting, validation, query building, and business logic hooks.
- Body-parser: This middleware is key to processing JSON requests, which is the backbone of RESTful API development.
- bcrypt.js: Security is paramount, and bcryptjs helps us to hash passwords securely before storing them in our database, ensuring that user credentials are protected.

- Nodemailer: Communication with users is facilitated through this module, which enables us to send emails directly from our server, such as password reset links or verification codes.
- Crypto: Built into Node.js, this module is used to perform cryptographic operations, including generating secure random strings that can be used for token generation.
- Path: This core Node.js module is used to work with file and directory paths, making it easy to manage file system paths across different operating systems.
- Session: We use the express-session middleware for handling user sessions, maintaining a secure and stateful interaction between our server and the client.

2 Codebase

Link to Github repository :- https://github.com/RushikeshChary/KnowYourCibo.git

Website URL :- https://knowyourcibo.onrender.com/

File Structure:

- 📁 /models:
  Here reside JavaScript files defining the structure of data models relevant to the food rating website. These models likely represent entities such as feedback, items (food items), restaurants, and users.
  - items.js
  - restaurants.js
  - user.js
- 📁 /public:
  This directory stores static assets such as images and client-side scripts and stylesheets. Considering this is a food rating website, the subdirectories may contain images related to user profiles, authentication, feedback, home page, search, and signup. Additionally, there might be JavaScript and CSS files responsible for styling and scripting different aspects of the website.
  - 📁 /EditprofileImages
  - 📁 /authenticationImages
  - 📁 /feedbackImages
  - 📁 /homeImages
  - 📁 /img
  - 📁 /profileImages
  - 📁 /searchImages
  - 📁 /signupImages
  - .DS_Store
  - D.jpg
  - editProfileScript.js
  - editProfileStyle.css
  - feedBackScript.js
  - feedBackStyle.css
  - forgot_script.js
  - forgot_style.css
  - hallScript.js
  - hallStyle.css
  - homeScript.js
  - homeStyle.css
  - img1.jpg

- loginStyle.css
- profileStyle.css
- restaurants.css
- restaurants.js
- script.js
- searchStyle.css
- signupStyle.css
- stylepage.css

📁 /views:
This directory contains the view templates rendered by the server and sent to clients. The views include pages such as restaurant listings, user profiles, feedback submission, login, signup, and possibly partials like footers. These templates are likely in EJS format.

- 📁 /partials
  - ❖ footer.ejs
- Restaurants.ejs
- editProfile.ejs
- feedback.ejs
- forgot_password.ejs
- hall.ejs
- home.ejs
- login.ejs
- profile.ejs
- searchPages.ejs
- signup.ejs

- .DS_Store

- .gitignore: A file specifying which files or directories should be ignored by Git version control.

- README.md: A markdown file containing information about the food rating website, such as how to set it up, its purpose, and possibly its features.

- app.js: This is the main entry point for the Node.js application, where the server is created and configured. It may also include middleware setup and other configurations.

- package-lock.json: This file contains detailed information about the exact versions of dependencies installed in the project, ensuring consistent builds across different environments.

- package.json: The metadata file for the Node.js project, containing information about the project, its dependencies, and scripts to run various tasks.

- server.js: This is a critical file in the application's architecture, responsible for setting up and configuring the server to handle incoming requests and manage the application's behaviour.

3 Completeness

1. **Home Page:** This page is the entry page of the website for all the users**.**
   • Login option - This is available here for those who have created a KnowYourCibo account and had to log in. For those who haven't, they have to sign up (option is made available on the login page).
   • Restaurants option -This is present as a button which on clicking redirects the user to the various types of restaurants available.
   • Feedback option - This is present as a button which on clicking redirects the user to the feedback page.
   • Profile option - This option is available for only those who have logged in. This is present as a button which on clicking redirects the user to their respective profile page.
   • Contact us section - This section is present as you scroll through and reach the bottom of the home page. Contacting details of the website developer are made available here for the users.

2. **Login Page:**
   •This web interface prompts users to authenticate themselves before gaining access to KnowYourCibo's luxuries.
   • Users need to input credentials such as email address and password for accessing their account.
   • Forgot password - If in case a user forgets his/her password,this option redirects them to the forgot password page.
   • The "Create new account?" option typically provides a link for new users to create an account. It redirects them to a signup page where they can input their information to sign up for access to the website.

3. **Signup Page:**
   • The signup page is where new users create an account by providing necessary information such as username, email address, password, and any additional required details.

   •The "Already have an account" option on this page typically provides a link or button for users who already possess an account. This option redirects users away from the signup process to the login section, where the users can login with their account credentials.

4. **Forgot Password:**
   • This page allows users who have forgotten their login credentials to initiate a password recovery process. It typically prompts users to enter their email address associated with their account to get an otp . Users need to input this otp for their account recovery.

5. Profile page:

• This page is accessible only if the user is logged in

• The profile page serves as a personalized space where users can view and manage their information, activities, and interactions within the KnowYourCibo website. It typically displays user details such as name, profile picture,email address, along with ratings, reviews and other contributions associated with the user's account.

• Personal favourites are shown (if any) on the right side section of the profile page.

•The edit profile button here redirects users to the edit profile page.

6. Edit Profile:

• The edit profile page allows users to modify and update their personal information and preferences within the website. Users can typically change their usernames and passwords. This page provides a means for users to keep their profiles accurate and up-to-date according to their evolving needs and preferences.

7. Functionality of Search:

• The search functionality enables users to quickly find restaurants, cuisines, or specific dishes based on their preferences.This enhances user experience by providing efficient and personalized access to a wide range of food options based on user preferences.

8. Restaurant Section:

• The restaurant section in home page serves as a central hub for users to discover, explore, and select from a wide range of dining establishments available on the KnowYourCibo platform.

• Clicking on a restaurant card expands to reveal more detailed information, including the restaurant's address, menu items with prices, customer rating and reviews, and photos.

9. Hall Page:

• A comprehensive menu featuring dishes offered by a particular restaurant, along with prices, ratings and reviews of a wide range of items help users browse and select their desired items.

• The "Popular Categories" section here enhances user convenience and satisfaction by offering a curated selection of trending food categories, making it easier for users to explore and order their desired cuisine with just a few clicks.

10. Feedback Form:

• The feedback page is a platform where users can provide comments, suggestions, or complaints regarding their experience with the website. It typically includes fields for users to input their feedback along with options to rate their satisfaction.

• The form is available on the homepage to all irrespective of them having an account.

• This form also requires users to input their name, email and date rendered.

Future Development Plans:

1. Reducing Response Time:

To mitigate the 4-5 second delay caused by the local database, consider implementing a caching mechanism to store frequently accessed data locally. Utilizing techniques like caching popular search results, menu items, or frequently visited pages can significantly reduce response time. Additionally, optimizing database queries and indexing can enhance database performance, thereby reducing latency. Explore cloud-based solutions for hosting the database to leverage scalability and faster access times.

2. Search History Functionality:

Introducing a search history feature allows users to easily access their past searches, enhancing user experience and engagement. Implementing this feature involves storing user search queries in a database along with timestamps. Displaying the search history prominently on the website interface and providing options for users to filter or refine their search based on past queries can further enhance usability. Additionally, offering personalized recommendations based on past searches can enrich the user experience.

3. Responsive Website Design:

Adopting a responsive web design approach ensures that the website adapts seamlessly to various devices, including mobile phones, laptops, and tablets, providing an optimal viewing experience. Utilize frameworks like Bootstrap or CSS Grid to create fluid layouts that adjust based on screen size. Prioritize mobile optimization by designing touch-friendly interfaces, optimizing images and media for faster loading, and ensuring intuitive navigation across different screen resolutions. Conduct thorough testing across multiple devices and screen sizes to validate responsiveness and identify areas for improvement.

4. User profile Enhancement :

We aim to enrich the user experience by allowing users to personalize their profiles with photos. This not only adds a personal touch but also helps users to identify themselves and others within the community or platform.Users will be able to upload their profile photos directly from their devices. We'll ensure a seamless and user-friendly interface for this process, supporting various file formats and providing clear instructions.

Appendix A - Group Log

Sr. No.	Date	Timings	Venue	Description
1	10/02/2024	3 pm - 5 pm	RM Building	Distributed work among the team and planned the implementation process.
2	26/02/2024	5 pm - 7 pm	RM Building	Discussed on development of back-end and brief details about the design of application.
3	05/03/2024	4 pm - 6 pm	RM Building	Everyone reported the progress of the work they were assigned and further work distribution was done.
4	09/03/2024	3 pm - 5 pm	RM Building	Discussed regarding implementation of the database and started the implementation.
5	12/03/2024	5 pm - 8 pm	RM Building	Discussed and resolved problems faced in integration of backend with frontend.started resolving them and added minor changes to CSS.
6	14/03/2024	4 pm - 9 pm	RM Building	Discussed and resolved problems faced in integration of backend with frontend. started resolving them. Started working on the Implementation document as well
7	17/03/2024	5pm - 8 pm	RM Building	Discussed and resolved problems faced in integration of backend with frontend. started resolving them and added some new features here and there.
8	18/03/2024	6pm - 9pm	RM Building	Finished the Software Implementation Document and resolved queries in application.