**Lab-3**

**19CSE461 -Net-Centric Programming**

**Name:** Maddipati Sandeep Chandra

**Roll No:** CB.EN.U4CSE19633

**Official Mail Id:** [cb.en.u4cse19633@cb.students.amrita.edu](mailto:cb.en.u4cse19633@cb.students.amrita.edu)

**Semester:** 7

**Application Name: Canteen Management System**

**About the Application:**

The application is a canteen website, which consists of various sections such as registration, sign in, menu, cart, myprofile, myoders, and the feedback of the site.

This is a food booking website which is used for ordering the food from a particular restaurant. In this website the user can order the food item that he is willing to have. But for ordering any food the user first need to be a logged in or should be a member of app. So a register form page will be opened at the start of application if the user is already a member there will be sign in option at the right end of the page. Once the user is logged in he will be directed to home page where there will be the description about the application. There will be links provided above so that the user can navigate to what ever page he want.

There will be an option called menu where the user can look at the menu and can order the food item into cart . From the cart the user can order food.

**Online URL:** <https://canteenmanagementsystem.netlify.app/>

**Modules in the application:**

Login

Register

Add food

View food

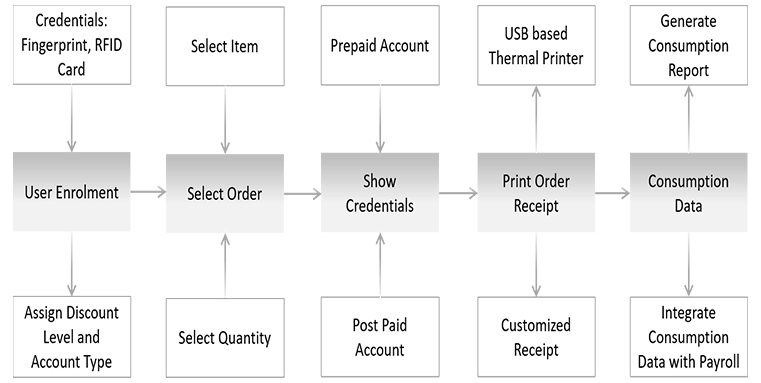
Order

my profile

feedback

Cart

**Provide a work Break Down Diagram for the application:**



**Write in your own words how the application has used Mean Stack Development**

MongoDB:

As a document database, MongoDB makes it easy to store structured or unstructured data. This format directly maps to native objects, so we don’t need to think about normalizing data.

ExpressJS:

Levee used Express JS as web application framework on top of the NodeJS. Since Express.js requires only JavaScript, it is easy to build the application in less time. This is the point where api interacts with the client.

Angular:

As angular is a component-based framework and single page client application, we used angular for frontend framework.

NodeJS:

We used for server-side programming, and primarily deployed for non-blocking, event-driven servers, such as traditional web sites and back-end API services.

**List of technologies/languages/framework/library used with purpose and version**

|  |  |  |
| --- | --- | --- |
| **technologies/languages/framework/library** | **Purpose** | **Version** |
| HTML | Describing the webpage structure, headings, text, tables, lists, photos, and hyperlinks | 5 |
| CSS | describing the presentation of Web pages, mobile responsive web page design and media queries and document styling | 3 |
| JS | To create dynamic and interactive web content, dynamic functionality and form validation | ES2015 |
| AngularJS | To html as template language, write client-side applications using JavaScript in a clean Model View Controller, cross platform and databinding features | 1.8.3 |
| Express’s | To write client-side applications using JavaScript in a clean Model View Controller, write client-side applications using JavaScript in a clean Model View Controller and efficient middleware | 4.18.2 |
| MongoDB | To manage document-oriented information, store or retrieve information. | 6.0.2 |
| TypeScript | To develop angular component modules | 4.9.3 |
| mongoose | NPM package to create data models and a simple Api for mongo dB | 6.7.2 |
| cors | To enable cross origin resource sharing | 2.8.5 |
| NodeJS | As an asynchronous event-driven JavaScript runtime, Node.js is designed to build scalable network applications | 16.14.1 |
| JSON | The standard medium of data transfer between server and client | 2.6.2 |

Write in your own words how the following concepts have been used in your application:

|  |  |
| --- | --- |
| HTML 5 | Basic Form structure with form input and controls (button, label, checkbox) |
| CSS | Styling of the front end and Fluid UI, media queries, mobile responsiveness |
| Angular JS | Managing the routing and various routes to be handled, data binding and dependency injection, routes for login, payment, checkout etc... have been added |
| Node JS | Running the server, scalable, asynchronous, concurrent, on local port or the process.env |
| Express JS | Single page server-side application and Rest API |
| XML | To represent the structured information of data |
| XML DTD |
| XML Schema |
| JSON | User data stored in JSON format for easier storage, access and mongo DB params maintained |
| MongoDB | Database connectivity, login and authentication, managing the user cart for tracking orders and payment |

|  |  |
| --- | --- |
| Number of Forms | 8 |
| Number of XML files | 3 |
| What is unique in your application? | Order categories and editing of purchased order, my profile, etc.. |
| Form Names for Evaluating the concepts of AngularJS, NodeJS, Express’s | Register, Add food, View food, Order, my profile, feedback |
| Tables for Evaluating the MongoDB concepts | Users, carts, feedback, orders, food |

Explanation of content in the Document in above headings: 5 Marks

For the following tables refer the concepts provided for learning from internet resources and fill the table

Weightage: 5 Marks

|  |  |  |
| --- | --- | --- |
| **Concept used from HTML5** | File Name | Additional Information |
| <body>, <image>, <audio>  Media elements  HTML slider, carousels | All Forms | Media elements |
| Tables | Products order and order summary in tabular format |
| <div>, <p>, Other structural HTML tags, flex box | Data storage and structural organization |
| Form controls | Forms for data entry and user input processing |
| Semantic elements (header, footer) | Contact, navigation bar |

Weightage: 10 Marks

|  |  |  |
| --- | --- | --- |
| **Concept used from XML** | File Name | Additional Information |
| XML | Signup, feedback, profile | To describe the data |
| XML DTD |
| XSD |
| Validation |
| CData,PCdata |

Weightage : 10 Marks

|  |  |  |
| --- | --- | --- |
| **Concept used from JavaScript** | File Name | Additional Information |
| DOM | index |  |
| Function calls | Nested function cals, for add item , remove item from cart, etc… |
| Event handling | Handle user input, actions, browser actions |
| Async Communication | To return the promise |
| Looping, conditional Statements | Retrieve and display orders in the cart, past orders |
| Promises |  |
| **Concept used from JSON** | File Name | Additional Information |
| Array | Userhome, adminhome, myprofile, viewcart | Data storage of the user from registration form like age, name etc in JSON int, string and user order purchase amounts of past orders in JSON list |
| List |
| Integer |
| Complex Data Structure |
| Arrow Function | Used in place of functions wherever possible |

Weightage : 20 Marks

|  |  |  |
| --- | --- | --- |
| **Concepts used from**  **Angular JS** | File Name | Additional Information |
| MVC | Index, oder history | HTML component rendering, Routing, single page application, and angular js validation, Logic and interaction of REST API’s |
| Filters, Tables |
| Scopes |
| Controller |
| Expression |
| Directives |

Weightage : 20 Marks

|  |  |  |
| --- | --- | --- |
| **Concepts used from**  **NodeJS** | File Name | Additional Information |
| Async/Await, dotenv | Index, myprofile, viewcart, feedback | To load ENV files into the application |
| http, file module, npm | http and file modules for http and working with the local file system |
| Event loop | To perform non blocking I/O operations |
| Call back | Calls multiple processes without waiting for return , adding product of one category and moving on t another category without waiting for first to return back |
| Promises |  |

Weightage : 20 Marks

|  |  |  |
| --- | --- | --- |
| **Concepts used from**  **ExpressJS** | File Name | Additional Information |
| Routing | Index | Routing requests for various API endpoints |
| HTTP methods, Middleware | Index | JSON parser, error handler, authentication middleware, static file storage |
| Env variables | index | Load up ENV variables into the application |
| Mongoose Models | models/ | Creating the data models |
| Cookie, Session | cart | Cookie and session info of order products, cart |
| Error Handling | Index, cart |  |

Weightage : 10 Marks

|  |  |  |
| --- | --- | --- |
| **Concepts used from**  **MongoDB** | File Name | Additional Information |
| Documents | Index, Userhome,  adminhome, myprofile, viewcart, cart | Main document under which different collections(tables), cart, user, etc… are maintained |
| Collections | Collection for the user |
| Create/Add to Databases | Newdb created for the user at registration and the corresponding records are inserted |
| Remove | Removing the already added item from the cart of the user |
| Sorting | Sorting items based on price |
| MongoDB Atlas | Used mongoose to connect to  MongoDB’s cloud database |

**Provide five features that can be added to application in future:**

1. Rating for food
2. Special Menu List
3. Bulk order Booking