**EXPOSYS DATA LABS**

**FULL STACK DEVELOPER PROJECT REPORT**

**BY**

**M.H.MOGDOOM KHAN SAHIB**

**IVth YEAR BE CSE**

**AALIM MUHAMMED SALEGH COLLEGE OF ENGINEERING**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.No** | **TITLE** | **PAGE. NO** |
| **1** | **ABSTRACT** | **3** |
| **2** | **INTRODUCTION** | **4** |
| **3** | **MANDATORY REQUIREMENTS** | **6** |
| **4** | **EXISTING METHOD** | **8** |
| **5** | **PROPOSED METHOD WITH ARCHITECTURE** | **10** |
| **6** | **METHODOLOGY** | **12** |
| **7** | **IMPLEMENTATION** | **14** |
| **8** | **CONCLUSION** | **16** |
|  | | |

**1.ABSTRACT**

**Abstract for Burger Customization Website:**

* "Brrrgrrr" is an online burger customization website that enables users to personalize their burgers by adding or removing ingredients. The website also allows users to create unique burgers from the available stock. Utilizing arrays, DOM manipulation, ES6 compatibility, and higher-order functions, the website provides an interactive interface for customizing and ordering burgers.

**Abstract for Blog Application:**

* *The Blog Application is an end-to-end platform that facilitates user registration, login, and blog post management, including addition, deletion, and updating of posts. The application also incorporates search functionality and offers the capability to save blog data in Excel or Word formats. Employing CRUD operations, file handling, and OOPs concepts, the application offers users a seamless experience in creating, managing, and searching for blog content***.**

**2. INTRODUCTION**

**Introduction for Burger Customization Website:**

* *Welcome to "Brrrgrrr" - your one-stop destination for mouthwatering burgers customized just the way you like them! In this fast-paced world, we understand the value of a quick, delicious, and personalized meal. Our online burger customization website aims to revolutionize the way you experience burgers.*
* *With "Brrrgrrr," you are not limited to pre-defined options. You have the freedom to choose from a wide range of ingredients and create a burger that perfectly suits your taste buds. Whether you're a meat lover, a vegetarian, or someone with unique taste preferences, we've got you covered.*
* *Our website leverages cutting-edge technologies, including arrays, DOM manipulation, ES6 compatibility, and higher-order functions, to ensure a seamless and user-friendly interface. So get ready to embark on a delightful journey of crafting your dream burger with "Brrrgrrr"!*

**Introduction for Blog Application:**

* *Welcome to our comprehensive Blog Application - your ultimate platform for expressing thoughts, ideas, and experiences through captivating blog posts. In this digital age, blogging has become a powerful means of communication and self-expression, and our application is designed to make this experience effortless and enjoyable.*
* *Our end-to-end blog application provides a range of functionalities, from user registration and login to creating, managing, and searching blog posts. With a user-friendly interface, you can easily unleash your creativity, share your stories, and connect with a global audience.*
* *Harnessing the power of CRUD operations, file saving capabilities in Excel or Word formats, and incorporating OOPs concepts, our application ensures smooth data management and organized code structure. Whether you're a seasoned blogger or just starting your blogging journey, our Blog Application caters to all your needs and empowers you to become a proficient blogger. Let your voice be heard with our Blog Application!*

**3. MANDATORY REQUIREMENTS**

**1) Arrays:**

* *You will need to utilize arrays to store and manage the list of available ingredients, as well as the customized burger options created by the users. Arrays can be used to hold data related to ingredients, such as names, prices, or availability.*

**2) DOM Manipulation:**

* *DOM (Document Object Model) manipulation is essential for dynamically updating the website's content based on user interactions. You will need to use JavaScript to manipulate the DOM elements, such as updating the burger ingredients or displaying the final customized burger.*

**3) ES6 Compatible:**

* *The project should be developed using JavaScript that is compatible with ECMAScript 6 (ES6) standards. ES6 introduced several new features and syntax improvements, such as arrow functions, template literals, and destructuring assignments. Utilizing these features can enhance the readability and maintainability of your code.*

**4) Higher-Order Functions:**

* *Higher-order functions are functions that can take other functions as arguments or return functions as results. In the context of this project, you can utilize higher-order functions to handle user interactions or perform operations on arrays of ingredients. For example, you might use higher-order functions like map, filter, or reduce to transform or process arrays of ingredients.*

**5) CRUD Operations:**

* *CRUD (Create, Read, Update, Delete) operations are essential for managing blog posts. You will need to implement functionalities to create new blog posts, retrieve existing posts, update posts, and delete posts. These operations will involve interacting with a database or data storage system to store and retrieve the blog post data.*

**6) File Saving using Excel/Word:**

* *The project requires the ability to save blog data in Excel or Word formats. You will need to implement functionality to convert the blog post data into the desired format (Excel or Word) and save it to a file. This can be done using appropriate libraries or APIs that support Excel or Word file manipulation.*

**7) OOPs Concepts:**

* *The project should be built using object-oriented programming (OOP) concepts. OOP allows you to organize your code into reusable and modular components. You will need to design and implement classes and objects that represent entities such as users, blog posts, and search functionalities. OOP principles such as encapsulation, inheritance, and polymorphism can help in creating a maintainable and scalable codebase.*

**4. EXISTING METHOD**

**Front-end Development:**

* *Burger Customization Website: Front-end development involves designing and creating the user interface of the website that users interact with directly. It includes implementing the user interface for customizing burgers, displaying available ingredients, handling user interactions, and showing the final customized burger. Front-end technologies like HTML, CSS, and JavaScript are used, and the mandatory elements (arrays, DOM manipulation, ES6 compatibility, and higher-order functions) are relevant to the front-end development.*

**Frontend Technologies:**

* *HTML, CSS, and JavaScript or a modern frontend framework like React, Angular, or Vue.js. Utilize ES6+ features for writing modern JavaScript code.*

**Backend Project (Blog Application):**

* *Objective: Build a blog application end-to-end with modules for login, sign up, adding, deleting, updating blog posts, search functionality, and file saving using Excel or Word.*

**Backend Technologies:**

* **Backend Framework:** *Choose a backend framework based on*

*your preferred language. Some popular options include Node.js with Express (JavaScript), Django (Python), Ruby on Rails (Ruby), or Spring Boot (Java).*

* **Database:** *Select an appropriate database system to store blog posts and user information. Options include MySQL, PostgreSQL, MongoDB, or SQLite.*
* **Object-Relational Mapping (ORM):** *If using a relational database, consider using an ORM like Sequelize (Node.js) or Django ORM (Python) to interact with the database using object-oriented concepts.*
* **User Authentication:** *Implement user authentication and authorization using techniques like JWT (JSON Web Tokens) or session-based authentication.*
* **File Handling:** *For saving blog posts in Excel or Word formats, you can use libraries or built-in modules specific to your backend framework and language.*

**5.PROPOSED METHOD WITH ARCHITECTURE**

* *The proposed method for both projects involves a client-server architecture, where the frontend serves as the client-side user interface, and the backend serves as the server-side logic and data processing. The backend communicates with the database to persist and retrieve data.*

**1. Frontend Project ( Burger Customization Website):**

**Method:**

* + *Utilize HTML, CSS, and JavaScript or a frontend framework (React, Angular, or Vue.js) to build the user interface.*
  + *Implement DOM manipulation to dynamically update the website based on user interactions, allowing them to customize their burgers.*
  + *Use arrays to store available ingredients and user-selected choices.*

**Architecture:**

* ***Frontend:*** *The frontend will be responsible for presenting the user interface, handling user interactions, and sending requests to the backend API for data processing.*
* ***Backend:*** *The backend will receive requests from the frontend, process the burger customization logic, calculate prices, and respond with the results.*
* ***Database:*** *Store the available ingredients and user preferences in a database to persist data between sessions.*

**2. Backend Project (Blog Application):**

**Method:**

* *Choose a backend framework (Node.js with Express, Django, Ruby on Rails, or Spring Boot) for server-side logic and HTTP handling.*
* *Use ORM (Sequelize, Django ORM, or Hibernate) to interact with the database and perform CRUD operations for blog posts.*
* *Implement user authentication (JWT or session-based) to handle user registration and login.*
* *Integrate search functionality for users to search and retrieve specific blog posts.*
* *Implement file handling to save blog posts in Excel or Word formats.*

**Architecture:**

* **Frontend:** *Develop a separate frontend (HTML, CSS, JavaScript, or a frontend framework) for the user interface, user registration, login, and displaying blog posts. It will communicate with the backend through RESTful APIs.*
* **Backend***: The backend will handle HTTP requests, authentication, blog post management, and search functionalities. It will communicate with the database for data storage.*
* **Database:** *Utilize a database system (MySQL, PostgreSQL, MongoDB) to store blog posts, user information, and other data.*

**6. METHODOLOGY**

**1. Frontend Project ( Burger Customization Website):**

**Project Planning and Requirements Gathering:**

*- Define the project scope and objectives.*

*- Gather detailed requirements, including burger customization options, ingredients, and user interface preferences.*

**Frontend Development:**

*- Set up the project using a frontend framework (React, Angular, or Vue.js) or vanilla JavaScript, HTML, and CSS.*

*- Implement the user interface based on the design, incorporating dynamic features for ingredient selection and burger customization.*

*- Utilize arrays to store the available ingredients and user-selected choices.*

**Testing and Debugging:**

*- Conduct thorough testing to ensure the website functions correctly and is bug-free.*

*- Perform cross-browser and cross-device testing to ensure compatibility.*

**Integration with Backend (Optional):**

*- If needed, integrate the frontend with the backend API to handle authentication or store user preferences.*

**Deployment:**

*- Deploy the frontend to a web server or hosting platform to make it accessible to users.*

**2. Backend Project (Blog Application):**

**Project Planning and Requirements Gathering:**

*- Define the project scope, features, and functionalities required for the blog application.Determine the technologies, database system, and backend framework to be used.*

**Database Design:**

*- Design the database schema to store blog posts, user information, and other relevant data.*

*- Choose an appropriate database system (relational or NoSQL) based on the project requirements.*

**Backend Development:**

*- Set up the backend project using the chosen backend framework (Node.js with Express, Django, Ruby on Rails, or Spring Boot).*

*- Implement user authentication using JWT or session-based authentication.*

*- Create endpoints to handle CRUD operations for blog posts and search functionality.*

**Database Integration:**

*- Integrate the backend with the database using an ORM (Sequelize, Django ORM, or Hibernate) or raw SQL queries.*

**Testing and Debugging:**

*- Conduct comprehensive testing to verify the backend's functionality, data handling, and API endpoints.*

*- Address and resolve any bugs or issues discovered during testing.*

**Deployment:**

*- Deploy the backend application to a server or cloud platform to make it accessible to the frontend and users.*

**Frontend Integration (Optional):**

*- If needed, integrate the backend API with the frontend to fetch blog posts, enable user registration, and login*

**7. IMPLEMENTATION**

**1. Frontend Project (Burger Customization Website):**

**Frontend Implementation:**

* *Set up the project structure and install the required dependencies using a frontend framework like React, Angular, or Vue.js, or use vanilla JavaScript, HTML, and CSS.*
* *Design and create the user interface, including the burger customization options, ingredient list, and order summary.*
* *Implement JavaScript functions to handle user interactions, such as selecting and customizing ingredients and dynamically updating the burger appearance and price.*
* *Use arrays to store the available ingredients and user-selected choices.*
* *Create components for the different sections of the website (header, ingredient list, customization options, order summary, etc.).*
* *Utilize DOM manipulation to dynamically update the content based on user interactions.*
* *Test the frontend thoroughly, ensuring that it functions as expected and is responsive across different devices.*

**2. Backend Project (Blog Application):**

**Backend Implementation:**

* *Choose a backend framework like Node.js with Express, Django, Ruby on Rails, or Spring Boot.*
* *Design and set up the database schema to store blog posts, user information, and other relevant data.*
* *Implement user authentication using JWT (JSON Web Tokens) or session-based authentication.*
* *Create API endpoints to handle CRUD operations for blog posts (Create, Read, Update, Delete) and search functionality.*
* *Integrate the backend with the database using an ORM (Sequelize, Django ORM, or Hibernate) or raw SQL queries.*
* *Implement file handling to save blog posts in Excel or Word formats, if required.*
* *Test the backend thoroughly, ensuring that it handles requests correctly and stores data securely in the database.*

**Frontend-Backend Integration:**

* *Connect the frontend to the backend API using HTTP requests (e.g., Axios or Fetch API).*
* *Implement user authentication on the frontend and ensure it communicates securely with the backend.*
* *Test the integration to ensure that data is retrieved and displayed correctly on the frontend.*

***Deployment:***

* *Deploy the frontend to a web server or hosting platform (e.g., Netlify, Vercel, GitHub Pages) to make it accessible to users.*
* *Deploy the backend to a server or cloud platform (e.g., Heroku, AWS, Azure) to make it accessible to the frontend and handle user requests.*

**8.CONCLUSION**

*In conclusion, both the Frontend ( Burger Customization Website) and Backend (Blog Application)For the Frontend project, the website provides users with a dynamic and interactive platform to customize their burgers. By implementing arrays, DOM manipulation, and ES6 compatible JavaScript, users can easily add or remove ingredients, leading to a personalized burger creation experience. Utilizing higher-order functions further enhances the efficiency of ingredient processing.On the other hand, the Backend project presents a comprehensive blog application with user authentication, CRUD operations, search functionality, and file saving capabilities. By incorporating CRUD operations, file handling, and OOPs concepts, the application efficiently manages blog posts and user interactions. Additionally, the implementation of user authentication ensures secure access to the application's features.Together, these projects demonstrate the essence of full-stack development, wherein the frontend and backend components work in harmony to provide a seamless user experience. Proper frontend-backend integration enables data flow, user registration, and efficient communication between the user interface and the server-side logic.Throughout the development process, adherence to best practices, code modularity, and comprehensive testing are essential for creating robust and reliable applications.*