1/18/2024

Muhammad Shaheer Kamran

SP22-BSE-072

Software Requirements Specifications

Web Technologies

Revive Men’s Lounge

**Revive Men's Lounge - Requirements and Specifications**

**1. Project Overview**

Revive Men's Lounge is a promotional website built on the MERN stack, with the primary goal of providing salon customers with a platform to view services, book appointments, and submit feedback. The project is developed using React and Vite for the frontend, MongoDB as the database, and Node.js with Express for the backend.

**2. Stakeholders**

* **Project Owner:** Muhammad Shaheer Kamran
* **Customers:** Visitors to the salon, potential customers
* **Developers:** Team involved in the development and maintenance of the website

***3. Functional Requirements***

**3.1 Customer Interaction**

* **View Services:**
  + Customers can browse and view the list of services offered by the salon.
* **Book Appointments:**
  + Users should be able to schedule appointments through the website.
* **Submit Complaints:**
  + A "Contact Us" component allows customers to submit complaints or feedback.

**3.2 User Authentication**

* **Secure Login:**
  + Implement user authentication for booking appointments and accessing personalized features.

**4. Non-functional Requirements**

**4.1 Performance**

* **Page Load Speed:**
  + Ensure fast loading times for a seamless user experience.

**4.2 Security**

* **Data Encryption:**
  + Implement encryption protocols to secure sensitive customer information.
* **Access Control:**
  + Define user roles and permissions to restrict access to certain features.

**4.3 Scalability**

* **Handle Traffic:**
  + Design the system to handle potential increases in user traffic.

**5. User Interface (UI) Design**

* **Frontend Framework:**
  + Utilize React with Vite for a responsive and interactive user interface.
* **CSS Framework:**
  + Tailwind CSS, with a focus on using DaisyUI library for styling.

**6. System Architecture**

* **Frontend:**
  + React and Vite for dynamic and fast UI.
* **Backend:**
  + Node.js and Express for server-side logic.
* **Database:**
  + MongoDB connected via Mongoose for data storage.

**7. Data Requirements**

* **Data Storage:**
  + Use MongoDB to store information about services, appointments, and customer feedback.

**8. Testing and Quality Assurance**

* **Testing Approach:**
  + Implement unit testing and integration testing for both frontend and backend components.

**9. Security Requirements**

* **Secure Data Transmission:**
  + Utilize encryption protocols for secure data transfer between frontend and backend.

**10. Deployment and Maintenance**

* **Deployment Strategy:**
  + Deploy the application on a secure server, ensuring continuous availability.
* **Maintenance:**
  + Regularly update dependencies and perform security audits.

**11. Documentation**

* **User Manuals:**
  + Provide documentation for users on how to navigate the website.
* **Developer Guides:**
  + Document codebase, API endpoints, and setup instructions.