assignment_category_12

Dear Candidates,

🎉 **Congratulations** on successfully clearing the first selection round! We were highly impressed by your application and skills. You are now invited to participate in the next stage of our selection process. This project will be your opportunity to demonstrate your problem-solving skills, showcase your creativity, and overcome challenges with innovative solutions. We're excited to see your unique talents in action and look forward to exceptional and high-quality work. This is your chance to shine and show your full potential. 🍀 Best of luck!

Project Theme: Service Review System

This project is a **Service Review Application System**, a platform designed to allow users to review and interact with services listed by others. Users can log in, add services, and share their experiences by posting reviews. They can also manage their reviews, and explore reviews of the services they've added. This system provides hands-on experience in building a full-stack application, integrating user authentication, managing CRUD operations, handling reviews, and ensuring secure database interactions.

Relevant website

User Capabilities:

✅ **Add/Update/Delete Services**: Users can manage services they have added, including editing details or removing them from the platform.

✅ **View Service Details**: Users can browse the list of available services, view detailed information, and explore reviews from other users.

✅ **Add, Edit, Delete Reviews**: Logged-in users can post reviews, update them, or delete them as necessary. Reviews will include ratings and textual feedback about the service.

✅ **Manage My Reviews**: Users can easily view and manage all the reviews they've posted through my reviews page.

Ensure the Following things to get 100% mark

- Include at least 15 meaningful commits on the client side & 8 meaningful commits on the server side with descriptive messages.
- Include a README file with the project name, purpose, live URL, key features, and any npm packages you have used.
- Ensure the website is fully responsive on mobile, tablet, and desktop.
- Secure Firebase configuration keys using environment variables.
- Secure your MongoDB credentials using the environment variable.
- Create a design that encourages recruiters. Color contrast should please the eye & ensure that the website has proper alignment, space and The website does not express gobindo design

Deployment Guideline

If your Deployment is not okay you will get 0 and may miss the chance of our upcoming rewards.

- Ensure that your server is working perfectly on production and not throwing any **CORS / 404 / 504** Errors.
- Ensure that your Live Link is working perfectly and that it is not showing errors on Landing in your system.
- ⚠️ ensure that the page doesn't throw any error on reloading from any routes.
- ⚠️ Add your domain for authorization to Firebase if you use **Netlify / surge**
- ⚠️ Logged in User must not redirect to Login on reloading any private route

By following these guidelines, your **Best Service Review Application System** will stand out with a professional and unique design. Good luck! 🚀

Important Restrictions

🚫 **No Similar Designs:**

Your project must be entirely original and must not resemble previous assignments such as Dragon News, Espresso Emporium, Genius Car, Career Hub Website, or any course module project.

Penalty Notice: Similar designs will result in a zero (0) grade.

Layout & Page Structure

Navbar:

- **Before Login:** Shows Logo, Home, Services, Login, and Register.
- **After Login:** Shows Logo, Home, Services, Add Service, My Reviews, My Services, User Avatar, and Logout Button.

<Dynamic main section/>

🎯Here is the main section which will show the dynamic page according to the route.

Footer:

- Website Logo/Name, Description, Copyright Information, and Useful Links.

Example of Main Layout Design

Header
Dynamic sections based on Routes
Footer

Home Page

The home page will be visible to all users and includes the following sections:

Banner Section:

- A simple image slider/carousel with at least three slides containing meaningful text highlights.

Featured Services Section:

- Show **6 service cards** using **MongoDB's** **limit()** **method**. Each card should display:
  - Service image, title, description, price, and a **"See Details"** Button.

🎯 When the user clicks the "See Details" button, they are redirected to the service details page.

- **Explore and implement any of the animations from the Framer Motion.**

Meet our partners section:

Create a **"Meet Our Partners"** section showcasing key partners or collaborators associated with your application. This section will be **static**, featuring partner names, logos, and brief descriptions of their contributions or services.

Extra Section

- Add 2 relevant and meaningful extra sections that are mentioned above on the Home page.

Authentication System

Login Page:

When you click the login button on the navbar it redirects to the login page. You have to use a password and email-based authentication to log in.

The login page will have-

- Email
- Password
- Google login/ GitHub- implement any of one
- A link that will redirect to the Register page

🎯**Here the email and password should match with the registered email and password. If it doesn't match, show an error message. You can show an error by using toast/sweet alert if you want.**

Register Page:

You have to use a password and email-based authentication to register. The Register page will have the following -

- Name
- Email
- photoURL
- password
- A Link that will redirect to the login page

- ❖ **For password validation you need to follow this -**
    - Must have an Uppercase letter in the password
    - Must have a Lowercase letter in the password
    - Length must be at least 6 character
- ❖ If any of this isn't fulfilled it will show an error message /toast
- ❖ After successful login or Register you need to show toast/sweet alert

🎯**Don't implement email verification or forget password method as it will inconvenience the examiner. If you want, you can add these after receiving the assignment result.**

Add Service Page: (Private Route)

This page allows logged-in users to create a service.

Fields to Fill:

- **Service Details:**
    - Service Image
    - Service Title
    - Company Name
    - Website
    - Description
    - Category
    - Price
    - Added date (Not as input)
    - userEmail (from auth)

What Happens Next:

- After clicking the **Add Service** button, show a success message and store the service in the database.

Service Page

This page displays all created services using cards. Each card will display essential service information, such as:

- Service Image
- Service Title
- Description
- Category
- Price

- **”See Details” Button**

Service Details Page

Displays detailed information about a specific service also show total review count and all reviews of the service.

Features:

- **Add Review:**
    - Textarea for writing text review
    - Rating Selection. You can use react-rating.
    - Review Posted Date (Not as input)
    - UserInfo (From Auth)
    - Add Review Button
- After submission, save the review in the database and display it on the UI. (User photo and name must be shown).

My Services Page: (Private Route)

This page displays services created by the logged-in user in **table format**.

Features:

- **Search:** Add an input field for search functionality.
- **Update Button:** Opens a form in a modal, allowing users to update the service information.
- **Delete Button:** Opens a confirmation modal to delete a service entry.

My Reviews Page: (Private Route)

Shows a list of reviews the logged-in user has submitted for in **one column card layout**. Display review Information on each card. Service Title, Text Review, Rating must be shown.

Features:

- **Update Button:** Opens a form in a modal where users can update their review details.
- **Delete Button:** Opens a confirmation modal to remove the review.

🎯 **Service Title** should be **read-only** field, ensure that users cannot modify it.

Additional

- **Dynamic Title:** Make your website title Dynamic. For every Route change, The Website Title will be changed based on that route.
- **404 page:** Add a 404 page/Not Found Page
- **Spinner:** Show a loading spinner when the data is in a loading state.
- **Toast:** For all the CRUD operations, show relevant toast/ notification/ sweet alert with a meaningful message

Challenge Requirement

Search Functionality: Implement a search feature that allows users to find services based on keywords (for example, title, category, company name). While searching can be done on the client side, it is recommended to implement it on the server for better performance and scalability.

Filter Functionality: Implement a filter feature on the **Services** page, allowing users to select a specific category from a dropdown menu (e.g., Food, Transport, IT). This functionality will enhance the browsing experience by enabling users to explore services tailored to their needs.

Countup: Shows how many users, reviews, and services are in the platform using the react-countup

JWT Authentication:

Login/Registration: Upon login or registration, the server should generate a JWT token and automatically set it in the client's cookies.

Securing APIs: Secure all POST, PATCH, and DELETE API routes (except JWT endpoints) by requiring the token for authorization. Unauthorized users should receive an error after login.

Error Handling (Optional): Handling error responses with an interceptor is optional but recommended for streamlined management of unauthorized errors and token expiration.

Optional Requirement (But Highly Recommended):

Implement any two tasks from the following optional list:

1. Try to use any other tailwind CSS library like - Material Tailwind, shadcn, chakra UI, flowbite.
2. Add a spinner when the data is in a loading state. You can add a gif/jpg, use any package or customize it using CSS.
3. Implement Pagination in the services page. Show 6-9 services per page.

What to Submit

1. Your client-side code GitHub repository
2. Your server-side code GitHub repository
3. Your live website link

Some Guidelines

1. Do not waste much time on the website idea. Just spend 15-20 minutes deciding, find a sample website, and start working on it.
2. Do not waste much time finding the right image. You can always start with a simple idea. Make the website and then add different images.
3. Don't look at the overall task list. Just take one task at a time and do it. Once it's done, pick the next task. If you get stuck on a particular task, move on to the next Task.
4. Stay calm, think before coding, and work sequentially. You will make it.
5. Be strategic about the electricity issue.
6. Use chatGPT to generate JSON data. You can use chatGPT for other purposes as well .