B12-A10_category-0017

Explanation Video 🎥 : [B12 A10 Freelance MarketPlace Requirement Explanation Video.mp4]

Project Name: Freelance MarketPlace

Project Theme

Build a full-stack web application where users can explore, add, update, and delete jobs or tasks in a digital platform. Authenticated users will have their own control to manage jobs or task data. Then, other users can agree to take the jobs or tasks. This project will demonstrate the student's ability to integrate a Node.js + Express.js backend, MongoDB database, and Firebase authentication into a single cohesive app.

Key Rules:

GitHub Commits:

- Minimum **15 notable commits** on the client-side.
- Minimum **8 notable commits** on the server-side.

Readme.md:

- Must include your **project name** and **live site URL**.
- Add a minimum of **five bullet points** highlighting your app's features.

Responsiveness:

- The app must be fully responsive for mobile, tablet, and desktop.

Environment Variables:

- Hide Firebase config keys and MongoDB credentials using environment variables.

Lorem Text:

- No Lorem ipsum text or default alert() usage allowed. Use toast notifications for success/error messages.

UI Design Requirements:

- Unique Design: First, decide what kind of website you want to make. Then, search online or check out websites like ThemeForest to get ideas for the design. But remember, your website idea shouldn't be similar to any projects you've done before or to any examples in our modules or conceptual sessions.
- You can also look for free resources on blogs to help with your website.

1. Keep the main heading style (font, size, color) consistent across all sections.
2.
3. Keep paragraph spacing balanced and text easily readable.
4. Maintain uniform image sizes and spacing.
5. Use the same button style as on the home page.
6. Ensure good spacing and proper alignment.
7. Navbar, Keep the heading/logo same style and size as on the home page.
8. Use a grid layout with equal image sizes.
9. Keep all cards equal height and width (especially in services, projects, or products section)
10. Use the new X logo instead of the old Twitter bird to match the latest rebrand
11. Responsiveness: Make it responsive for all devices, including mobile, tablet, and desktop views.

Resources:

- https://uiverse.io/
- https://devmeetsdevs.com/
- https://bootcamp.uxdesign.cc/free-images-and-resources-collection-for-website-c77f2fc46ce5

- https://themeforest.net/?srsltid=AfmBOopTj6PNz51iuV2YJXUtBP8nt19_zT5LG2dToAjIHQqzNCzregn0
- https://codecanyon.net/?srsltid=AfmBOooRoUfeK7IOROpchCuA4hPVj5P9WRmtDQJ9K0E6Yhf4VTrHhXKt

Hosting:

- Use **Netlify / Surge / Firebase** for the client-side and **Vercel** for the server-side.
- Ensure no route reload errors occur.
- Add your deployment domain for Firebase authorization.
- Logged-in users should **not be redirected to Login** upon refreshing a private route.

Main Requirements

Layout Structure

- Include a **Navbar** , **Footer**, and **Main Content Area**.
- The **Navbar** must have navigation links: Home, All Jobs, Add a Job, My Accepted Tasks, and Login/Register.

Conditional Rendering:

- If the user is not logged in, show "Login" and "Register".

- If the user is logged in, show their photoURL, displayName on hover, and a "LogOut" button.

- The main section will be changed based on route.
- The **Footer** must contain copyright text.

Home

- Create a visually appealing homepage with:
  - **Banner:**
    Add a elegant banner with **animation, buttons (Mentioning how reliable the marketplace platform, Create a Job)**
  - **Dynamic Section:**
    Display the latest **6 Jobs** added to the platform, fetched dynamically from the MongoDB database.
  - **Two Static Sections (your choice):**
    Example ideas:
    - "Top Categories" – display Job categories with images
    - "About The Platform (your platform name)" – short intro about the platform

---

Authentication

User Login:

- Create a Login page with a form containing:
  Email, Password, Forget Password (text only), and Login button.

- On successful login, navigate the user to the home page or their desired route.

- On failure, show a toast/error message.

- Add links for “Register” and a **Google Login button**.

- On successful Google login, navigate to the desired route/home page.

User Registration:

- Create a registration page with a form containing:
  Name, Email, Photo URL, Password, and Register button.

- Password validation rules:

  - Must include at least one uppercase letter

  - Must include at least one lowercase letter

  - Must be at least 6 characters long

- On successful registration, navigate to the home page.

- On failure, show a toast/error message.

- Include a link for “Login” and a **Google Login button**.

💡 **Do not** implement email verification or forget password for this milestone.

CRUD Operation

Routes:

- `/addJob` → Page to add a new job

- `/allJobs` → Page to view all jobs from the database

- `/allJobs/:id` → Page to view a single job details

- `/updateJob/:id` → Page to update a single job/task info

- `/deleteJob/:id` → Page to delete a single job

- `/myAddedJobs` → Page showing only jobs added by the logged-in user

- `/my-accepted-tasks` → Page showing jobs that will be done by the logged-in user

Data Structure (stored in MongoDB):
Each book should contain:

{

```
  "title": "Job Title",

  "postedBy": "Job Post creator’s Name",

  "category": "Web Development / Digital Marketing / Graphics Designing etc.",

  "summary": "Short description of the book",

  "coverImage": "imgbb image URL",

  "userEmail": "email of the user who added the job"

}
```

Functional Requirements:

- Users can **Add** new jobs (with image url).

- Users can **Read** all jobs posted on the “All Jobs” page.

- Users can **Update** their own posted jobs (title, category, summary, cover image).

- Users can **Delete** their posted jobs.

- Data must be stored and fetched from MongoDB Atlas.

All Jobs Page:

- Display all jobs in a table / grid format. You can take inspiration from LinkedIn or any Freelance marketplace to show all the jobs as such.

- Show key info such as title, category, posted By, summary, cover image

- Include a "View Details" button to go to the job / task detail page (`/allJobs/:id`).

View Details Page:

- Show full details of the job in a nicely designed format. Take inspiration from any Freelance marketplace website.
- Allow users to accept the job by pressing the “Accept” button, those accepted jobs will appear in the `/my-accepted-tasks` page.
- This will be a private route.

Add Job Page:

Create a form to add data to the database with the following fields:

- Title
- Posted By (already filled from Logged-In user’s displayName)
- Category (make it a dropdown menu)
- Summary (Text Area field)
- **coverImage (Image Url)**
- User Email (already filled from Logged-In user’s email),
- Posted Date / Time (already filled automatic after posting the data on database, no need to fill this field from UI)

This will be a private route.
After successful submission, show a success message using any npm packages such as react-toastify, react-hot-toast or sonner etc.

🚩🚩🚩Updated:

Note: Using browser alert is not allowed, You can add Image Url for CoverImage

---

Update Page:

- Create a pre-filled form with all editable fields (title, category, summary, cover image).

- After submission, show a confirmation message upon successful update using any npm package.

My Accepted Tasks

- In this page, show accepted jobs from the job details page in Tabular or Grid format

- In each row / grid-card , there will be DONE and a CANCEL button (could be ✅ and ❌ icons)

- Clicking DONE or CANCEL button will remove the row / grid card from the UI instantly and from the database as well

Other Requirements

- Show a **Loading Spinner** when data is fetching.

- Create a **Custom 404 Error Page** for invalid routes.

- Use **TanStack Query** or **Axios** for data fetching.

- Protect routes like
  - `/addJob`,
  - `/myPostedJobs`,

- `/updateJob/:id`
- `/deleteJob/:id`
- `/allJobs/:id`
- `/my-accepted-tasks`

so only authenticated users can access them.

Challenges

1. **Advanced Filtering:**
   On the All Jobs page, implement sort functionality based on **Posted Date/Time**. (Hint for sorting)

2. Add a **dark/light theme toggle** on the homepage.

3. Don't allow accepting those jobs posted by the logged-in user. Only allow accepting jobs posted by other than logged-in users.

4. **Firebase Admin sdk authorization**

5. Explore these packages and implement at least one:

   1. Motion / Framer Motion

   2. React Bits

   3. GSAP

What to Submit

- **Client-side GitHub repository link**
- **Server-side GitHub repository link**
- **Live website link**