

# Project Requirements

A company is looking for a full-stack web developer capable of creating a simple full-stack website similar to online marketplaces.

Make sure your website design is unique. Visit ThemeForest, Dribbble, google, etc., to get some ideas. You can explore tailwind component libraries.

## Main Requirements

1. Focus on making the website visually appealing. Ensure that:
  - Color contrast is pleasing to the eye
  - The website has proper alignment and space
  - If needed, customize the design of any component you are taking from any component library. (For example, you are using daisy ui & have taken a card component from Daisy, if needed, customize the styling of the card to make it reasonable rather than just copy & paste it.)
2. Make sure to keep the navbar and footer on all the pages except on the 404-page. Create a reasonable and meaningful footer. (including website logo+name, copyright, some contact information, social media links, address, etc.)
3. You should have a navbar with the name+logo, Home, Add job, My posted jobs, My Bids, Bid Requests, User profile picture and user name and Register/login.
4. **Login & Registration systems:** On the Registration and Login pages, display relevant error messages when necessary

**Login Page:** When a user clicks on the login button, they will be redirected to the login page having the following:

- Email/Password
- Google Sign-in
- A link that will redirect to the registration page

**Registration Page:** The Registration page will have the Email/Password form having the following fields:

- Name
- Email
- Password
- PhotoURL

**Note:** Do not enforce the email verification method and forget & reset password method.

## 5. Home page

- **Banner**
  - Banner section will have a banner/carousel (images should be relevant to the website)
- **Browse By Category**
  - The website will feature a tab-based system for browsing jobs, representing the categories of web development, digital marketing, and graphics design. Each tab will display a minimum of four **cards**.
  - These job data will come from mongodb database. Users will add job data from the Add Job page. [See No. 7]
  - Implement tabs using this NPM package: [React-tabs](#)

**Note:** In each card, you will have to show the following information:

- Job title
- Deadline
- Price range
- Short description
- Bid now Button / Or make the whole card clickable.

- Added extra two sections relevant to the website . (Optional - Highly Recommended)

## 6. Job details page:

After clicking on the bid now button or card, user will be redirected to the job details route (/job/:id) containing the information (Name, Deadline, Price Range, Description etc) and “**Place A Bid**” form section.

**Note:** The “**Place A Bid**” form section will have the following input fields:

- Price (your bidding amount)
- Deadline
- Email [read-only: get it from user context]: (Who is bidding)
- Buyer Email [read-only: get it from user context]: (The job owner who has posted this job.)
- Bid on the project button (This button will be disabled for the owner who posted this job)

**Note:** When the 'Bid on the project' button is clicked, data will be stored in a MongoDB database. Once the data is saved, show a toast [don't use javascript alert] and after that user will be redirected to the **My Bids** page. [See No. 9 for detail information about **My Bids** page]

**Note:** Don't let the employer bid on his own job.

## 7. Add jobs: Create an Add Jobs page where there will be a form having the following fields

- Email of the employer(read-only)
- Job title
- Deadline
- Description
- Category
  - These are the names of tab options of the homepage.
  - Implement a dropdown for selecting categories.
- Minimum price
- Maximum price

- Add job button

**Note:** When the '**Add Job**' button is clicked, data will be stored in a MongoDB database. Once the data is saved, show a toast [don't use javascript alert] and after that user will be redirected to the **My Posted Jobs** page.[See No. 8]

8. **My posted jobs:** If a user logs in, they will see the My posted jobs page which will show all the jobs information this user added from the **Add job** page in tabular format. Each table row will have an **update** and **delete** button.

- Update action- If they click the update button, they can update the job information
  - email(not editable)
  - Job title
  - Deadline
  - Description
  - Category
  - Minimum price
  - Maximum price
  - Update button

**Note:** you can show the update form in a modal or in another route. When the '**update button**' is clicked, data will be updated . Once the data is updated, show a toast [don't javascript alert] and redirect user back to My Posted jobs page.

- Delete action- After clicking the delete button, the job will be removed. Before the delete, ask for a delete confirmation.

**Note:** If a user logs in they will only see the jobs they have added. The user cannot see the jobs other users added.

**\*\*Don't Let a user Update/Delete other employers job posts.\*\***

9. **My bids:** Create a My bids page where you will show all the bid information this logged in user added from the job details page in the tabular form. Each row of the table will have the following information:

- Job title
- Email
- Deadline
- Status
- Complete button(conditional)

**Note:** initially status is **Rending**.If the job owner rejects the bid then status will be **Rejected**.

Or if the bid is accepted by the job owner (who posted this job) status is **In Progress**. If the status is **In Progress** the complete button is enabled. After clicking the **complete button**, status is **complete** and the complete button will be disabled.

10.**Bid Requests:** job owner will see all the bid requests made by other users on their posted jobs in tabular format in this page. Job owners can manage (accept/reject) job status in this page. Each row of the table will have the following information:

- Job title
- Email (who bid the job)
- Deadline
- Price
- Category
- status
- Accept button
- Reject button

**Note: Note:** Initially accept button and reject button will be enabled. After clicking the Reject button, the status will be changed to rejected and the reject button will be disabled. If the user clicks on the Accept button, status will be updated to **In Progress** and the accept button will be disabled.

**11. 404 page:** Create a 404 page. Add any interesting jpg/gif on the 404 page. Do not add header & footer on this page. Just add a jpg/ gif & a **Back to home** button. The Back to home button will redirect the user to the home page.

12. Use the Environment variable to hide the Firebase config keys and MongoDB credentials.

**13. Private Route:**

- My bids page
- Add job page
- My posted jobs
- Bid Request page
- Job detail page

**Optional (But Highly Recommended)**

**Make the deadline strict:**

1. When adding a job, take the deadline input as a valid date.
2. If a job's deadline is crossed (compare using js Date.now()), no one can place a bid on it. It means the "Bid on this project" button will be disabled based on the deadline.