Full Stack Development with MERN

Project Documentation

1. Introduction

- **Project Title: Fliexihub** - Freelance Platform
- **Team Members:**
  - **L.G.MATHESH -** Fullstack Developer
  - **J.JOHN BENISH-** FrontEnd Developer
  - **J.DEEPAN CHAKRAVARTHI-** UI/UX and BackEnd Developer
  - **S.J.DHINAGAR**- Database and Content Manager

2. Project Overview

- **Purpose:** The purpose of the **Freelancer Hub** project is to create a platform that facilitates the connection between clients and freelancers, enabling efficient and secure project posting, bidding, and collaboration. This platform is designed to streamline the freelancing process by providing a user-friendly interface for clients to post projects, review freelancer profiles, and select suitable candidates. For freelancers, it offers a straightforward application process, project submission, and feedback mechanisms.

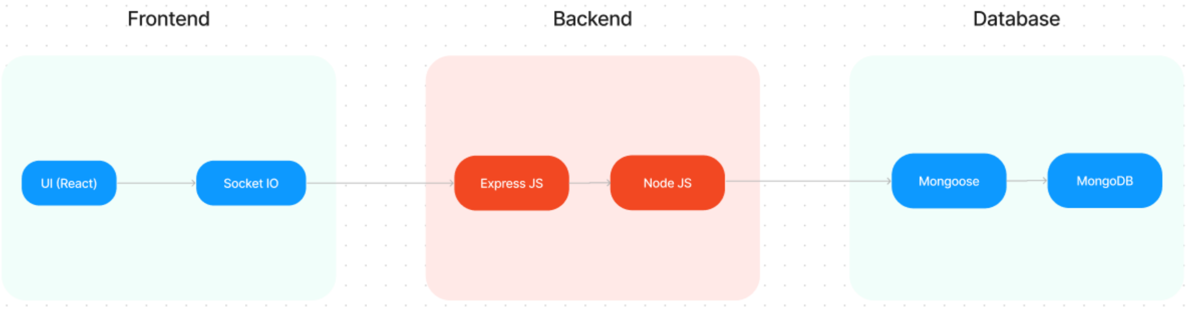
- The platform also features an admin team responsible for ensuring data security, policy enforcement, and conflict resolution, fostering a reliable and professional environment for both clients and freelancers. Through real-time communication, structured workflows, and a dedicated admin support system, Freelancer Hub aims to become the preferred platform for clients and freelancers seeking trustworthy collaboration in various fields
- **Features:**
  - User Registration and Login (Freelancers and Clients)
  - Profile creation and management
  - Project posting and bidding functionality
  - Messaging system for project communication
  - Payment system integration (optional)
  - Search and filtering of freelancers and projects

3. Architecture

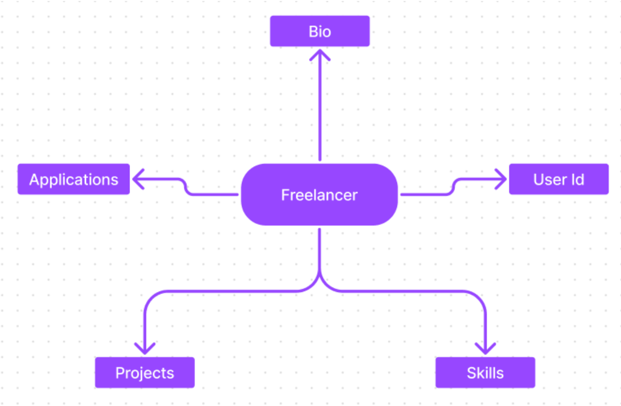
TECHNICAL ARCHITECTURE

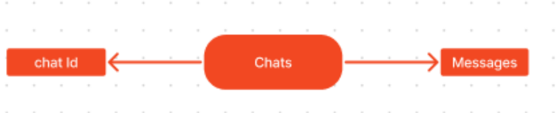
- The technical architecture of Freelancer Hub follows a client-server model, where the frontend serves as the client and the backend acts as the server. The frontend encompasses the user interface, presentation, and integrates the Axios library to facilitate easy communication with the backend through RESTful APIs.
- To enhance the user experience, the frontend leverages the Bootstrap and Material UI libraries, creating a real-time and visually appealing interface for users.

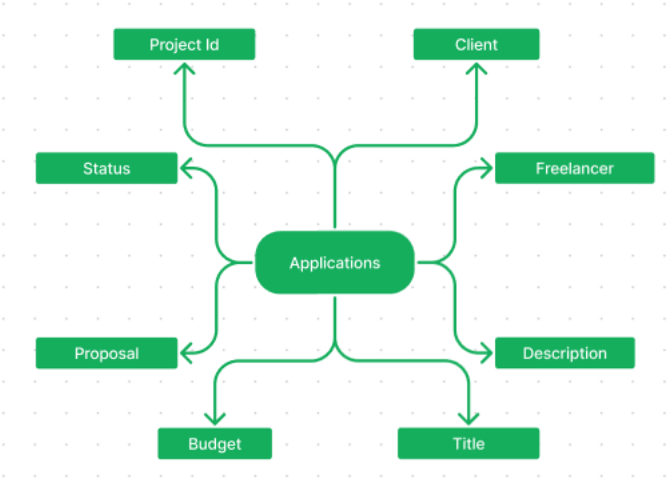
- On the backend, we utilize the Express Js framework to manage server-side logic and communication. Express Js provides a robust foundation for handling requests and responses efficiently.
- For data storage and retrieval, Freelancer Hub relies on MongoDB. MongoDB offers a scalable and efficient solution for storing various data, including user-contributed locations and images. This ensures quick and reliable access to the information needed to enrich the local tourism experience.
- In conjunction, the frontend and backend components, complemented by Express Js, and MongoDB, together form a comprehensive technical architecture for Freelancer Hub. This architecture facilitates real-time communication, efficient data exchange, and seamless integration, ensuring a smooth and immersive experience for users contributing to and exploring their local surroundings.

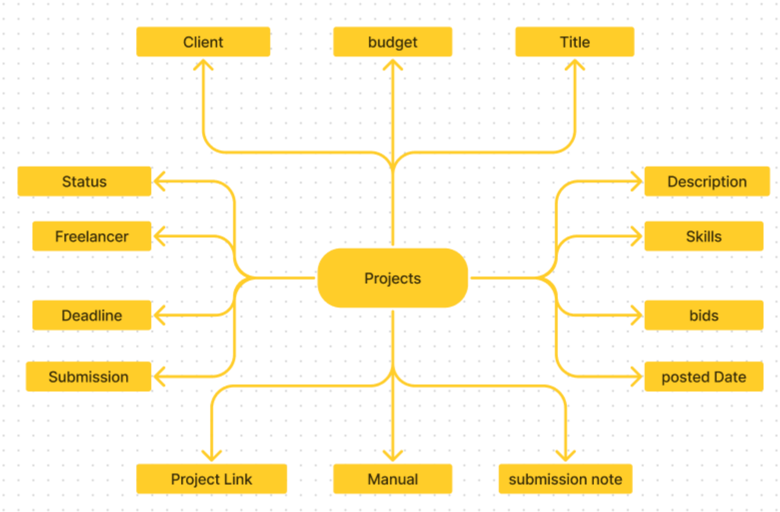


ER DIAGRAM









- Freelancer Hub connects clients with skilled freelancers through a user-friendly platform. Clients can post projects with details and browse freelancer profiles to find the perfect match. Freelancers can submit proposals, collaborate with clients through secure chat, and securely submit work for review and payment. An admin team ensures quality and communication, making Freelancer Hub a go-to platform for both clients and freelancers.

4. Setup Instructions

- **Prerequisites:**
  - Node.js (v12+)
  - MongoDB (local or cloud instance like MongoDB Atlas)
  - Git (for version control)

- **Installation:**
  1. Clone the repository:

     ```
     bash

     git clone https://github.com/yourusername/FliexiHub.git
     ```

  2. Navigate to project root.
  3. Install dependencies for both frontend and backend:

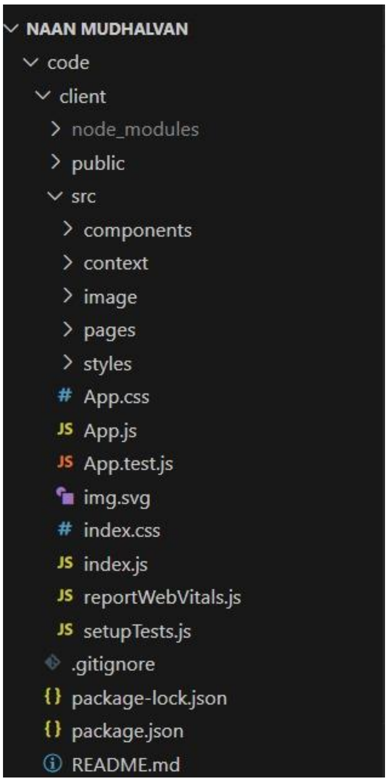
     ```
     bash

     cd client
     npm install
     cd ../server
     npm install
     ```

  4. Configure environment variables:
     - Set up `.env` file in the `server` folder with necessary details like **MongoDB URI, JWT secret, etc.**

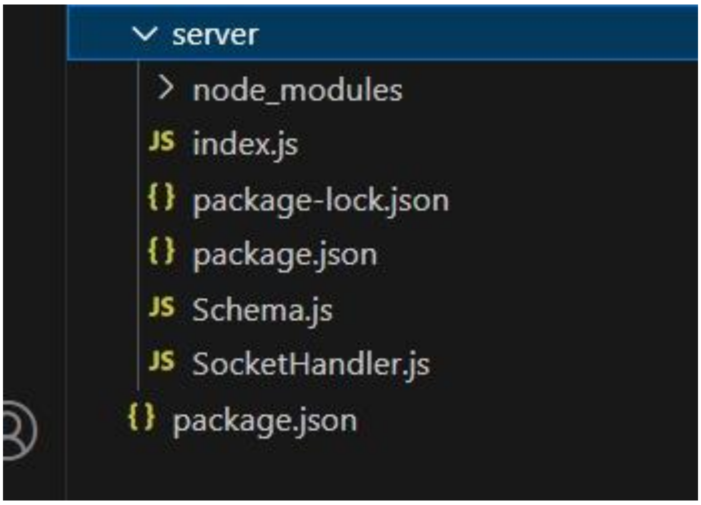
5. Folder Structure

- **Client:**
  The React application is organized into components, pages, services, and assets directories.



```
⌄ NAAN MUDHALVAN
  ⌄ code
    ⌄ client
      > node_modules
      > public
      ⌄ src
        > components
        > context
        > image
        > pages
        > styles
        # App.css
        JS App.js
        JS App.test.js
        img.svg
        # index.css
        JS index.js
        JS reportWebVitals.js
        JS setupTests.js
      .gitignore
      {} package-lock.json
      {} package.json
      ⓘ README.md
```

- **Server:**
  The Node.js backend follows a structured approach with routers, controllers, models, and middlewares.



```
server
  node_modules
  index.js
  package-lock.json
  package.json
  Schema.js
  SocketHandler.js
package.json
```

6. Running the Application

To run the Freelancer Hub application locally, you need to start both the frontend and backend servers. Follow these steps:

1. Install Dependencies:

- Navigate to the project’s root directory and run:

```
cd client
npm install
cd ../server
npm install
```

2.Start the Development Server:

- **Frontend:**
  Start the frontend server by navigating to the `client` directory and running:

```
sql

npm start
```

- **Backend:**
  Start the backend server by navigating to the `server` directory and running:

```
sql

npm start
```

7. API Documentation

1. User Registration

- Endpoint: `POST /api/auth/register`
- Description: Registers a new user (client or freelancer) on the platform.
- Request Body:

```json
{
 "username": "string",
 "email": "string",
 "password": "string",
 "role": "client or freelancer"
 }
```

- Response:

```json
{
"message": "User registered successfully",
"user": { "id": "string", "username": "string", "role": "client or freelancer" }
}
```

2. User Login

- Endpoint: `POST /api/auth/login`
- Description: Authenticates a user and returns a JWT token.
- Request Body:

```json
{
"email": "string",
"password": "string"
}
```

- Response:

```json
{
"token": "JWT token",
"user": { "id": "string", "username": "string", "role": "client or freelancer" }
}
```

3. Get User Profile

- Endpoint: `GET /api/users/:userId`
- Description: Retrieves user profile information, including portfolios and reviews.
- Response:

```json
{
  "id": "string",
  "username": "string",
  "email": "string",
  "role": "string",
  "portfolio": [/* array of projects or samples */],
  "rating": "number"
```

  }

4. Update User Profile

- Endpoint: `PUT /api/users/:userId`

- Description: Updates user profile details.

- Request Body (example):

  ```json
  {
    "bio": "string",
    "skills": ["string"],
    "portfolio": ["url"],
    "rating": "number"
  }
  ```

5. Post a New Project

- Endpoint: `POST /api/projects`

- Description: Allows clients to post a new project.

- Request Body:

  ```json
  {
    "title": "string",
    "description": "string",
    "budget": "number",
    "deadline": "date",
    "skillsRequired": ["string"]
  ```

  }
  ~~~
- Response:
  ```json
  {
    "id": "string",
    "message": "Project created successfully"
  }
  ~~~

6. Browse Projects

- Endpoint: `GET /api/projects`
- Description: Retrieves a list of all active projects.
- Query Parameters (optional):
  - `search`: Filter projects by keywords or skills.
- Response:
  ```json
  [
    {
      "id": "string",
      "title": "string",
      "description": "string",
      "budget": "number",
      "skillsRequired": ["string"],
      "deadline": "date"
    }
  ```

  ]

7. Get Project Details

 - Endpoint: `GET /api/projects/:projectId`

 - Description: Retrieves detailed information about a specific project.

 - Response:

  ```json

  {

    "id": "string",

    "title": "string",

    "description": "string",

    "budget": "number",

    "skillsRequired": ["string"],

    "deadline": "date",

    "client": { "id": "string", "username": "string" }

  }

  ```

Applications (Bidding)

8. Submit Proposal for Project

 - **Endpoint**: `POST /api/projects/:projectId/applications`

 - Description: Allows freelancers to apply for a project by submitting a proposal.

 - Request Body:

  ```json

  {

    "freelancerId": "string",
  ```

```
  "proposal": "string",
  "rate": "number"
}
```

- Response:

```json
{
  "message": "Application submitted successfully",
  "applicationId": "string"
}
```

9. Get Applications for a Project

- **Endpoint**: `GET /api/projects/:projectId/applications`

- Description: Retrieves all proposals submitted for a specific project (client view).

- Response:

```json
[
  {
    "applicationId": "string",
    "freelancer": { "id": "string", "username": "string", "rating": "number" },
    "proposal": "string",
    "rate": "number"
  }
]
```

Chat and Collaboration

10. Send Message in Project Chat

- **Endpoint**: `POST /api/projects/:projectId/chat`

- Description: Allows clients and freelancers to communicate within a project.

- Request Body:

```json
{
  "senderId": "string",
  "message": "string"
}
```

- Response:

```json
{
  "message": "Message sent successfully",
  "chatId": "string"
}
```

11. Get Chat Messages for Project

- **Endpoint**: `GET /api/projects/:projectId/chat`

- Description: Retrieves all chat messages for a specific project.

- Response:

```json
[
  {
```

  "sender": { "id": "string", "username": "string" },

  "message": "string",

  "timestamp": "date"

 }

]

Admin Functionality (Optional)

12. Admin: Manage Users

- **Endpoint**: `GET /api/admin/users`

- Description: Allows admins to retrieve a list of users for monitoring purposes.

- Response:

```json

[

 { "id": "string", "username": "string", "role": "client or freelancer", "status": "active or suspended" }

]

```

13. Admin: Manage Projects

- **Endpoint**: `GET /api/admin/projects`

- Description: Allows admins to monitor project listings.

- Response:

```json

[

 { "id": "string", "title": "string", "client": { "id": "string", "username": "string" }, "status": "open or closed" }

## 8. Authentication
```

- **Authentication & Authorization:**
  Authentication is handled via **JWT (JSON Web Token)**. Tokens are issued upon login and used to authenticate API requests. Authorization ensures only authorized users can access certain endpoints, e.g., only freelancers can apply for jobs, and only clients can post jobs.

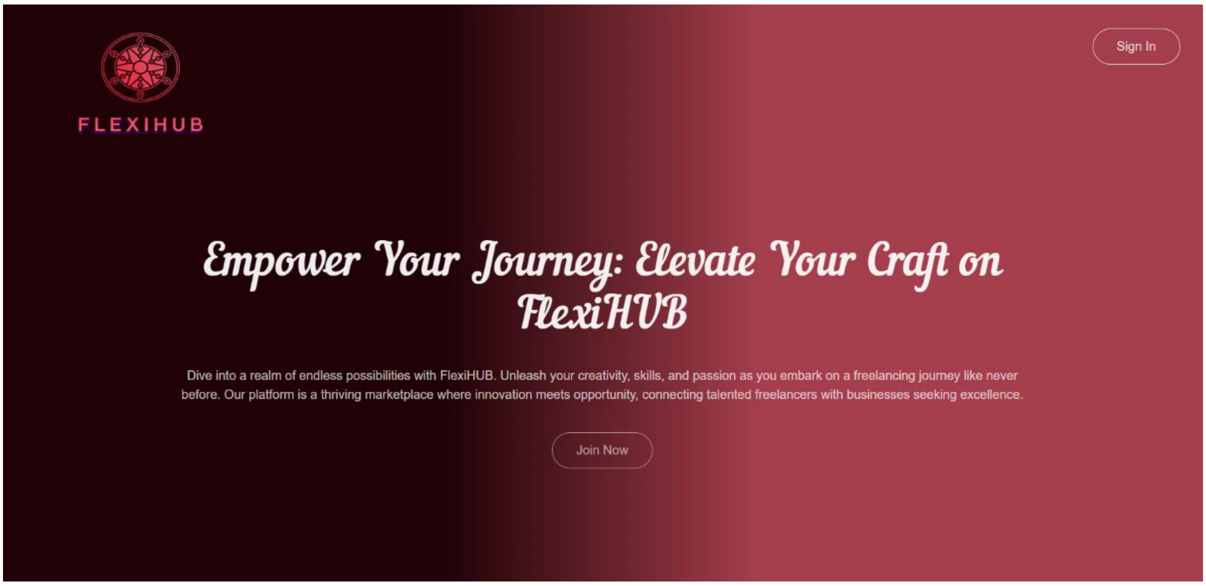
9. User Interface

- Designing reusable UI components like buttons, forms, and project cards.
- Defining the layout and styling for a visually appealing and consistent interface.
- Implementing navigation elements for intuitive movement between features.
- These steps will create a user-friendly experience for both freelancers and clients.

On completing the development part, we then run the application one last time to verify all the functionalities and look for any bugs in it. The user interface of the application looks a bit like the images provided below

- **UI Features:**

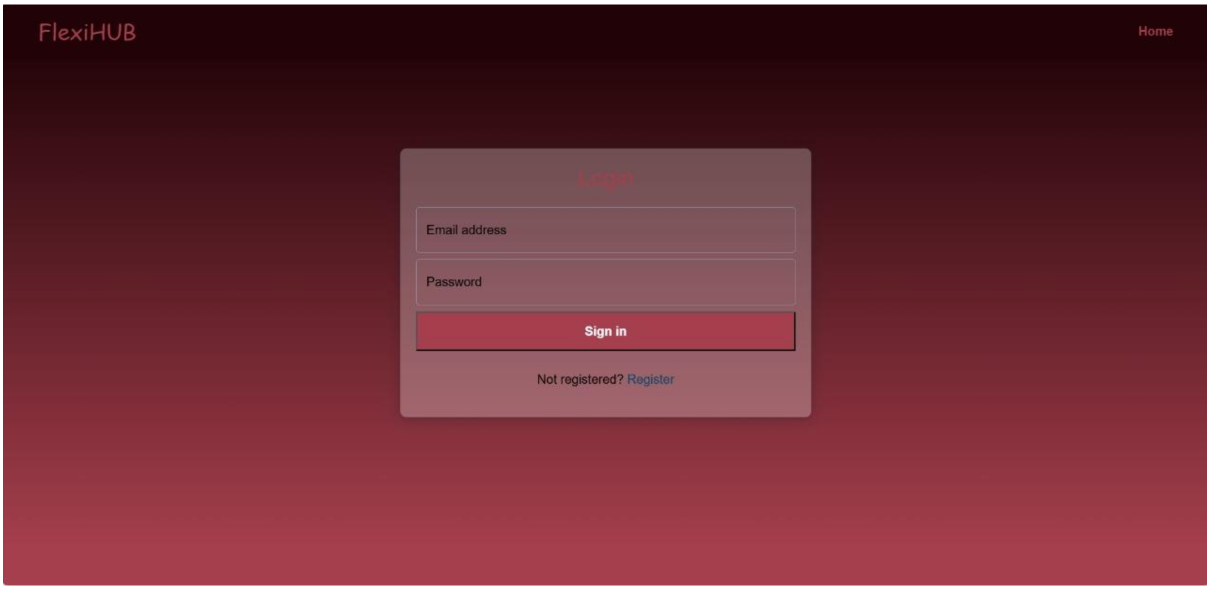
- **Landing page:**



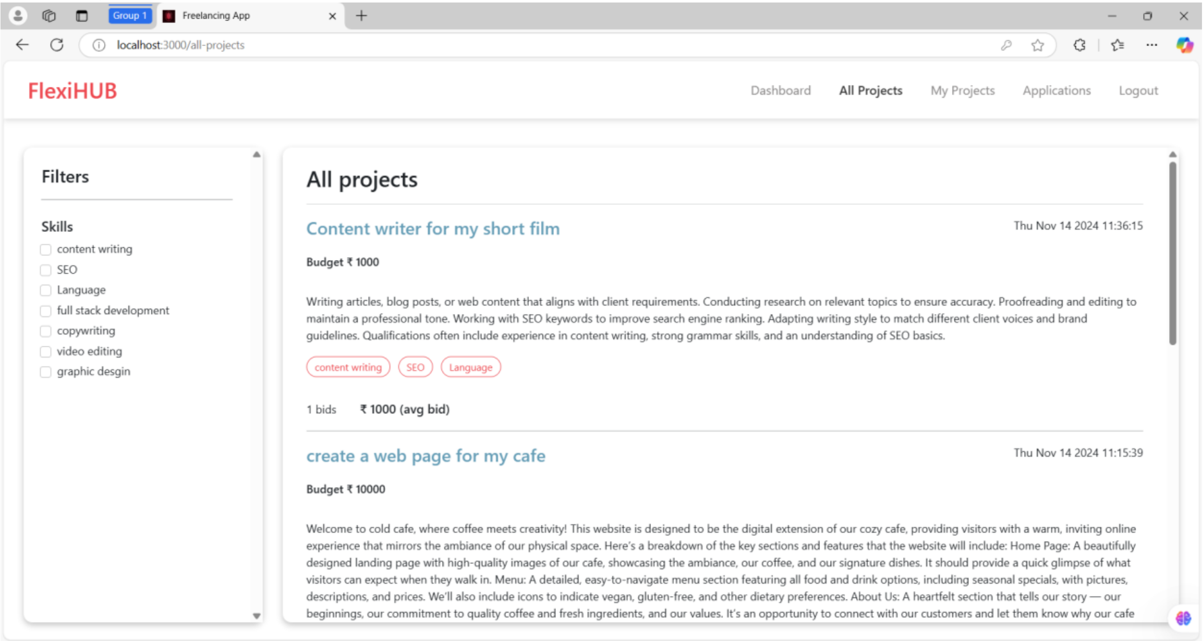
Empower Your Journey: Elevate Your Craft on FlexiHVB

Dive into a realm of endless possibilities with FlexiHUB. Unleash your creativity, skills, and passion as you embark on a freelancing journey like never before. Our platform is a thriving marketplace where innovation meets opportunity, connecting talented freelancers with businesses seeking excellence.

Authentication:



All projects:



Filters

Skills

- □ content writing
- □ SEO
- □ Language
- □ full stack development
- □ copywriting
- □ video editing
- □ graphic desgin

All projects

Content writer for my short film

Thu Nov 14 2024 11:36:15

Budget ₹ 1000

Writing articles, blog posts, or web content that aligns with client requirements. Conducting research on relevant topics to ensure accuracy. Proofreading and editing to maintain a professional tone. Working with SEO keywords to improve search engine ranking. Adapting writing style to match different client voices and brand guidelines. Qualifications often include experience in content writing, strong grammar skills, and an understanding of SEO basics.

content writing SEO Language

1 bids **₹ 1000 (avg bid)**

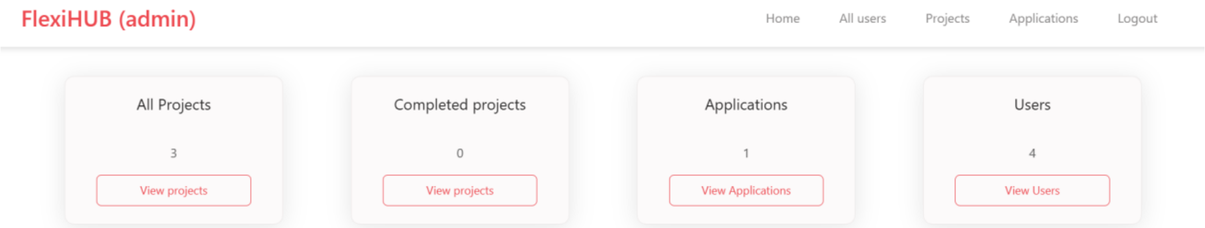
create a web page for my cafe

Thu Nov 14 2024 11:15:39

Budget ₹ 10000

Welcome to cold cafe, where coffee meets creativity! This website is designed to be the digital extension of our cozy cafe, providing visitors with a warm, inviting online experience that mirrors the ambiance of our physical space. Here’s a breakdown of the key sections and features that the website will include: Home Page: A beautifully designed landing page with high-quality images of our cafe, showcasing the ambiance, our coffee, and our signature dishes. It should provide a quick glimpse of what visitors can expect when they walk in. Menu: A detailed, easy-to-navigate menu section featuring all food and drink options, including seasonal specials, with pictures, descriptions, and prices. We’ll also include icons to indicate vegan, gluten-free, and other dietary preferences. About Us: A heartfelt section that tells our story — our beginnings, our commitment to quality coffee and fresh ingredients, and our values. It’s an opportunity to connect with our customers and let them know why our cafe

Admin dashboard:



All Projects

3

Completed projects

0

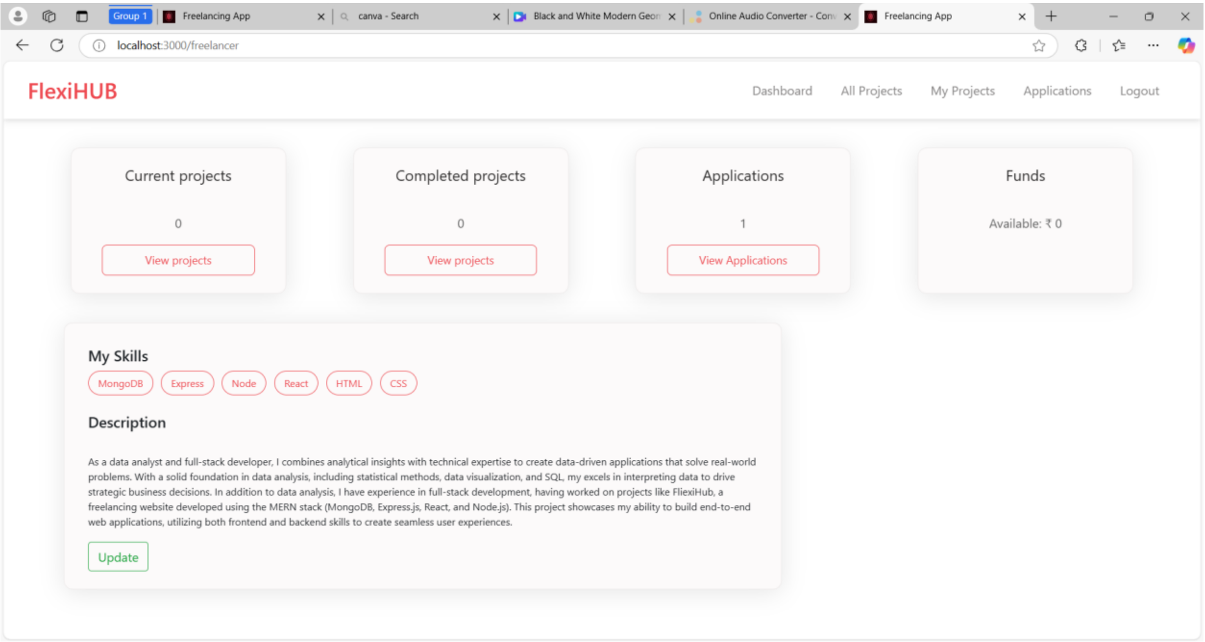
Applications

1

Users

4

Freelancer dashboard:



Current projects

0

Completed projects

0

Applications

1

Funds

Available: ₹ 0

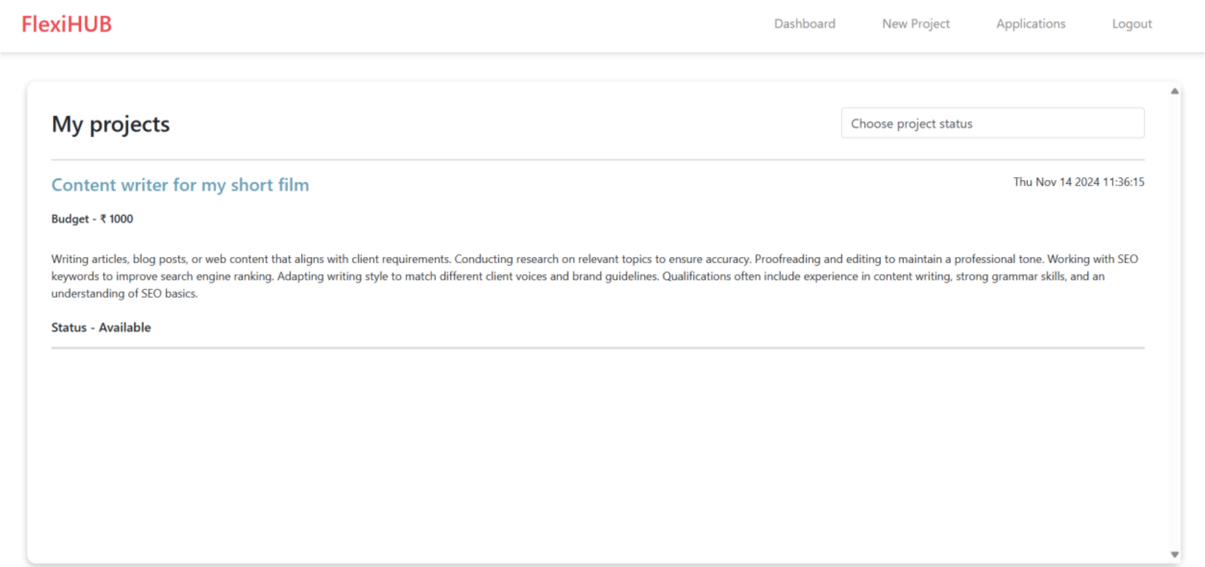
My Skills

MongoDB Express Node React HTML CSS

Description

As a data analyst and full-stack developer, I combines analytical insights with technical expertise to create data-driven applications that solve real-world problems. With a solid foundation in data analysis, including statistical methods, data visualization, and SQL, my excels in interpreting data to drive strategic business decisions. In addition to data analysis, I have experience in full-stack development, having worked on projects like FlexiHub, a freelancing website developed using the MERN stack (MongoDB, Express.js, React, and Node.js). This project showcases my ability to build end-to-end web applications, utilizing both frontend and backend skills to create seamless user experiences.

Freelance projects:



My projects

Choose project status

Content writer for my short film

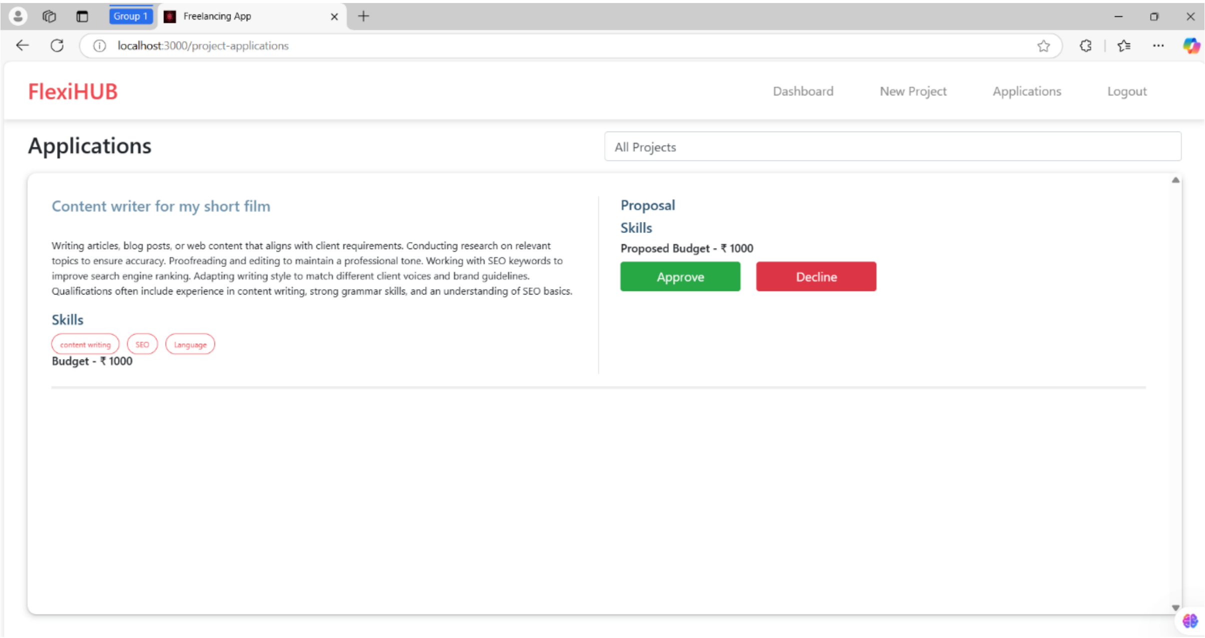
Thu Nov 14 2024 11:36:15

Budget - ₹ 1000

Writing articles, blog posts, or web content that aligns with client requirements. Conducting research on relevant topics to ensure accuracy. Proofreading and editing to maintain a professional tone. Working with SEO keywords to improve search engine ranking. Adapting writing style to match different client voices and brand guidelines. Qualifications often include experience in content writing, strong grammar skills, and an understanding of SEO basics.

Status - Available

Applications:



Applications

All Projects

Content writer for my short film

Writing articles, blog posts, or web content that aligns with client requirements. Conducting research on relevant topics to ensure accuracy. Proofreading and editing to maintain a professional tone. Working with SEO keywords to improve search engine ranking. Adapting writing style to match different client voices and brand guidelines. Qualifications often include experience in content writing, strong grammar skills, and an understanding of SEO basics.

Skills

content writing · SEO · Language

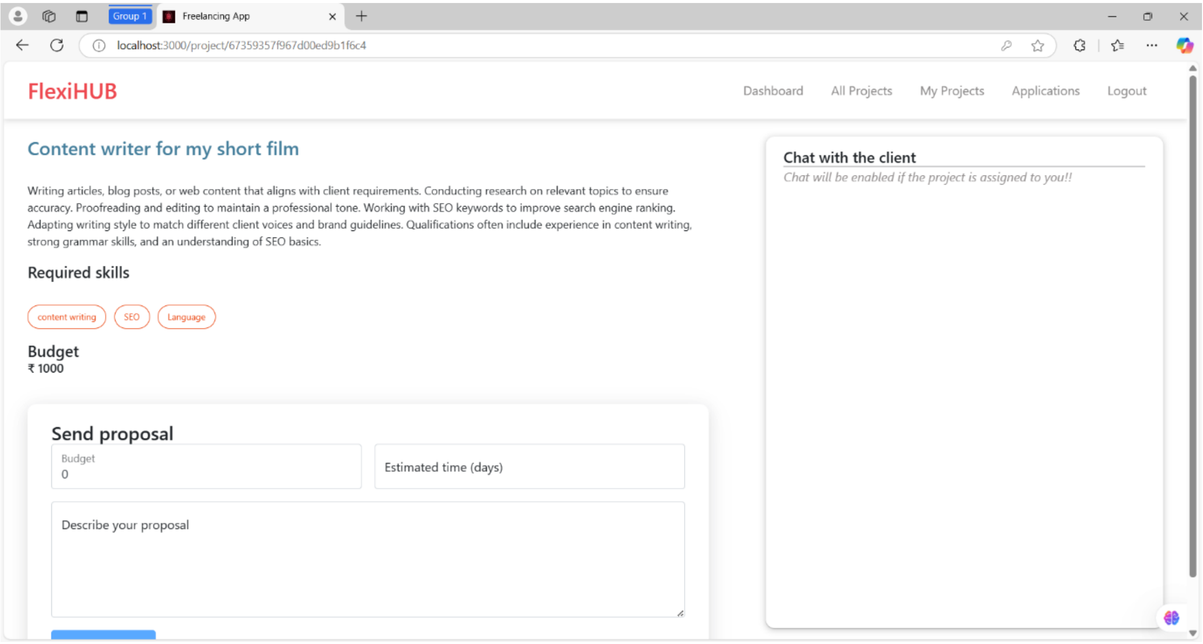
Budget - ₹ 1000

Proposal

Skills

Proposed Budget - ₹ 1000

Project page:



Content writer for my short film

Writing articles, blog posts, or web content that aligns with client requirements. Conducting research on relevant topics to ensure accuracy. Proofreading and editing to maintain a professional tone. Working with SEO keywords to improve search engine ranking. Adapting writing style to match different client voices and brand guidelines. Qualifications often include experience in content writing, strong grammar skills, and an understanding of SEO basics.

Required skills

content writing · SEO · Language

Budget
₹ 1000

Send proposal

Budget: 0

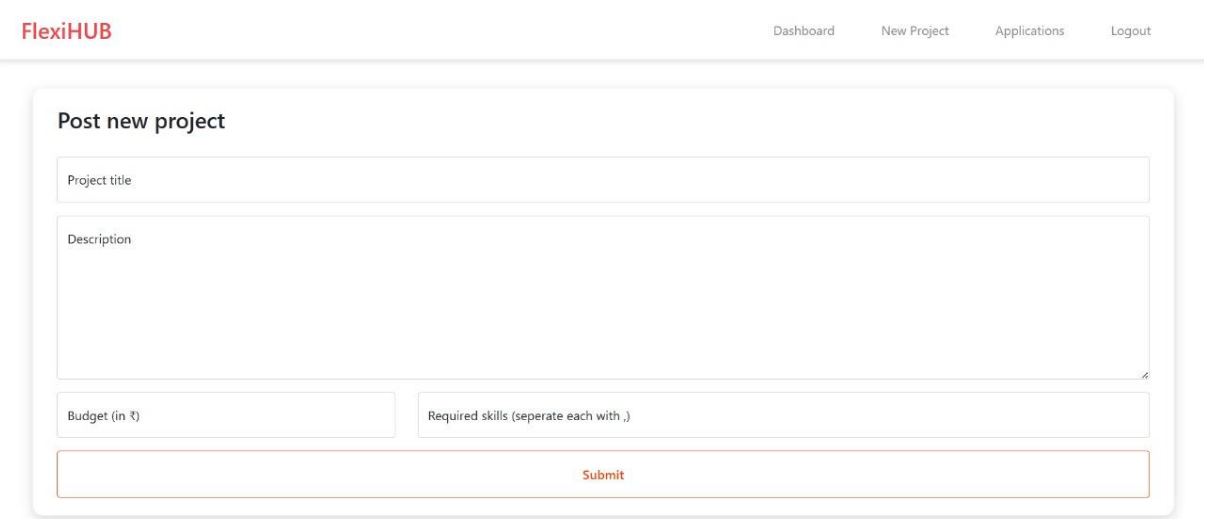
Estimated time (days)

Describe your proposal

Chat with the client

Chat will be enabled if the project is assigned to you!!

New project:



Post new project

Project title

Description

Budget (in ₹)

Required skills (seperate each with ,)

10. Testing

Testing Strategy and Tools

1. Unit Testing

• Objective: Test individual components and functions to ensure they work as expected in isolation.

• Tools: Jest and Mocha for JavaScript testing, particularly with backend logic and React components.

2. Integration Testing

• Objective: Verify that different modules or services within the system interact and work together correctly.

• Tools: Super test for testing API endpoints, allowing verification of the interactions between frontend and backend.

3. End-to-End (E2E) Testing

• Objective: Test the application's entire workflow from a user's perspective to validate the complete user experience.

• Tools: Cypress and Selenium for simulating real-world interactions, covering use cases such as project posting, bidding, chat, and payment.

4. User Interface (UI) Testing

• Objective: Ensure the frontend behaves as expected and is visually consistent across different devices and browsers.

• Tools: Cypress and Jest with React Testing Library for UI testing, along with Browser Stack for cross-browser and mobile device compatibility.

5. Performance Testing

• Objective: Assess the platform's speed, stability, and scalability, especially under high user load scenarios.

• Tools: JMeter and Apache Bench for load testing, identifying areas where the server or database might need optimization.

6. Security Testing

• Objective: Detect and address vulnerabilities in user authentication, data protection, and communication protocols.

- Tools: OWASP ZAP and Postman for vulnerability scanning and testing API security, as well as verifying data encryption on the backend.

7. Regression Testing

- Objective: Ensure that new updates or code changes do not negatively impact existing functionality.

- Tools: Automated testing scripts with Jest or Cypress, covering core features like registration, project management, and messaging.

8. Usability Testing

- Objective: Evaluate the user experience by collecting feedback from real users to make the interface intuitive and easy to use.

- Tools: Survey tools like Google Forms or User Testing to gather feedback from testers, focusing on navigation ease, visual design, and interaction clarity.

9. Acceptance Testing

- Objective: Ensure the system meets the requirements and behaves as expected for real user workflows.

- Tools: Manual testing performed by QA testers or end-users, with predefined test cases covering all major functionalities to confirm user requirements are fulfilled.

11. Github link

https://github.com/Johnbenish258/freelancer_website/commit/baec16163f98ac480f4efaf9d5bea39c5c5c7f08

12. Known Issues

- **Profile Loading Delays**: Some users may experience delays in loading profile data when viewing multiple profiles consecutively. This issue likely stems from server response times during peak usage.
- **Chat Synchronization**: Occasionally, messages in the project chat may not sync in real-time. This requires a refresh to view the latest messages.
- **Mobile Responsiveness**: Certain pages, like the project listing and profile review, may not display optimally on smaller screens and may require further CSS adjustments.
- **File Attachment Limitations**: Large files may occasionally fail to upload in the chat system, especially if they exceed the size limit set on the server.

13. Future Enhancements

• **Advanced Search and Filtering**: Implement a more comprehensive search feature, allowing users to filter projects by criteria like budget range, deadline, and freelancer ratings.

• **Project Milestones and Payments**: Introduce a milestone-based payment system where clients can release payments upon the completion of predefined project stages, enhancing flexibility and security for both parties.

• **Enhanced Mobile App Experience**: Develop a mobile application to improve usability on mobile devices, ensuring smooth navigation and functionality.

• **AI-Based Skill Matching**: Add an AI feature that matches clients' project requirements with freelancers' skills and ratings to improve the project matching process.

• **Portfolio Building Tools for Freelancers**: Provide freelancers with tools to create detailed portfolios, including client testimonials and work samples, directly on the platform.

• **Multi-Language Support**: Expand the platform's reach by adding support for multiple languages, making it accessible to a broader audience globally.