

FREELANCING WEBSITE USING MERN

Introduction

- **Project Title:** WORKIFY
- **Team ID (NM2024TMID01849)**
- **Members:**
 1. Naveenkumar P
 2. Mohammed Moosha S
 3. Dhivakar V
 4. Shigin K S
 5. Dhanu R

Project Overview

- **Purpose:** WORKIFY is an innovative freelancing platform that bridges the gap between clients in need of specific project-based work and freelancers seeking opportunities to showcase and monetize their skills. The platform is designed to make the project workflow seamless from posting and applying for projects to completing them and processing payments. Workify aims to create a transparent and efficient ecosystem

that benefits both clients and freelancers, facilitating smooth communication and project management.

- **Goals:**

- To offer clients an intuitive and efficient way to post projects with detailed requirements and budget constraints.
- To provide freelancers with a straightforward method to browse, apply for, and manage projects.
- To ensure project completion and payment processes are secure and user-friendly.
- To empower admins with tools to monitor the platform's activity and manage user interactions effectively.

- **User Roles:**

- **Client Dashboard:** Clients can post new projects, view ongoing and completed projects, approve or reject project submissions, and communicate with freelancers via the integrated chat feature.
- **Freelancer Dashboard:** Freelancers can browse available projects, submit proposals with detailed bids, interact with clients through messages, and submit completed project work for approval.
- **Admin Panel:** The admin role provides oversight capabilities, allowing monitoring of user registrations, project postings,

submitted proposals, and chat interactions. Admins can manage and oversee the platform to ensure smooth operations.

- **Project Lifecycle:**

- Clients post projects specifying requirements and budget.
- Freelancers browse projects and submit proposals for consideration.
- Clients review proposals, approve a suitable freelancer, and initiate the project.
- Freelancers submit completed work, including a project link and description, for client review.
- Upon client approval of the submitted work, funds are transferred to the freelancer.

- **Chat Functionality:**

- Real-time chat feature to enhance communication between clients and freelancers, facilitating discussions, clarifications, and project updates.

Architecture

· Frontend

Ø Framework: React.js

- **Components:** The frontend is designed using reusable React components that represent different parts of the application, such as user dashboards, project forms, chat interfaces, and notification panels.
- **Routing:** React Router is utilized for client-side routing, allowing smooth navigation between pages such as the client dashboard, freelancer project listings, and admin panel.
- **State Management:** The application leverages React's useState and useContext hooks for local and global state management, while more complex data handling is managed using third-party libraries like Redux (if applicable).
- **Styling:** The UI is styled using CSS-in-JS libraries (e.g., styled-components) or traditional CSS for custom and responsive design, providing a seamless experience across devices.
- **Security:** Prevents unauthorized access to specific pages with protected routes, verified through token-based authentication.

- **Backend**

- Ø **Framework:** Node.js with Express.js

- **Structure:** The backend is organized into controllers, routes, and middleware:

- § **Controllers:** Contain the business logic for handling HTTP requests, such as creating projects, handling proposals, and managing user interactions.

- § **Routes:** Define the endpoints and link them to the appropriate controller functions.

- § **Middleware:** Custom middleware ensures secure user sessions, validates JSON Web Tokens (JWT), and processes errors.

- **Services:** Service modules separate complex operations, such as payment processing or notification handling, from the main controller logic.
 - **Real-Time Communication:** Integrated using libraries like Socket.io for the chat feature, enabling real-time messaging between clients and freelancers.

- **Database**

- Ø **Database System:** MongoDB

- **Schema Design:**

- § **Users Collection:** Stores user profiles with information like role (client, freelancer, admin), name, email, hashed passwords, and profile details.

- § **Projects Collection:** Contains project details, including the client ID, project description, budget, status (e.g., open, in progress, completed), and associated freelancer proposals.

- § **Proposals Collection:** Captures the proposals submitted by freelancers, including freelancer ID, project ID, bid details, and proposal status.

- § **Chats Collection:** Manages chat data, including sender and receiver IDs, timestamps, and messages.

- **Relationships:**

- § One-to-many relationship between users and projects (a client can have multiple projects).

- § One-to-many relationship between projects and proposals (a project can have multiple freelancer proposals).

- **Indexes:** Critical fields, such as user email and project IDs, are indexed to improve query performance.

Setup Instructions

· Prerequisites

Before setting up the project, ensure you have the following installed:

- **Node.js:** Version 14.x or higher for running JavaScript code outside the browser.
- **MongoDB:** Either a Community Server or a cloud-based MongoDB Atlas account.
- **Git:** For cloning the repository.

· Installation Steps

1. Clone the GitHub Repository:

- Open a terminal or command prompt and navigate to your desired directory.
- Run the following command to clone the repository: `git clone [repository URL]`

- Replace [repository URL] with the actual link to your GitHub repository.

2. Navigate to the Project Directory:

- Change to the root directory of the cloned project: cd workify

3. Install Dependencies:

- The project contains separate folders for the client (frontend) and server (backend).
- Open two terminal windows or command prompts and follow these steps:

For the Client:

§ Navigate to the client folder: cd client

§ Install all Dependencies: npm install

For the Client:

§ Navigate to the client folder: cd server

§ Install all Dependencies: npm install

4. Modify Database Connection:

- Open the index.js file in the server directory.
- Update the MONGO_URI to point to your MongoDB Connection string (local or Altas).

FOLDER STRUCTURE

```
client
  > node_modules
  > public
  < src
    < components
      Login.jsx
      Navbar.jsx
      Register.jsx
    > context
    > images
    > pages
    < styles
      admin
      client
      freelancer
      # authenticate.css
      # landing.css
      # navbar.css
      # App.css
      JS App.js
      JS App.test.js
      # index.css
      JS index.js
    & logo.svg
```

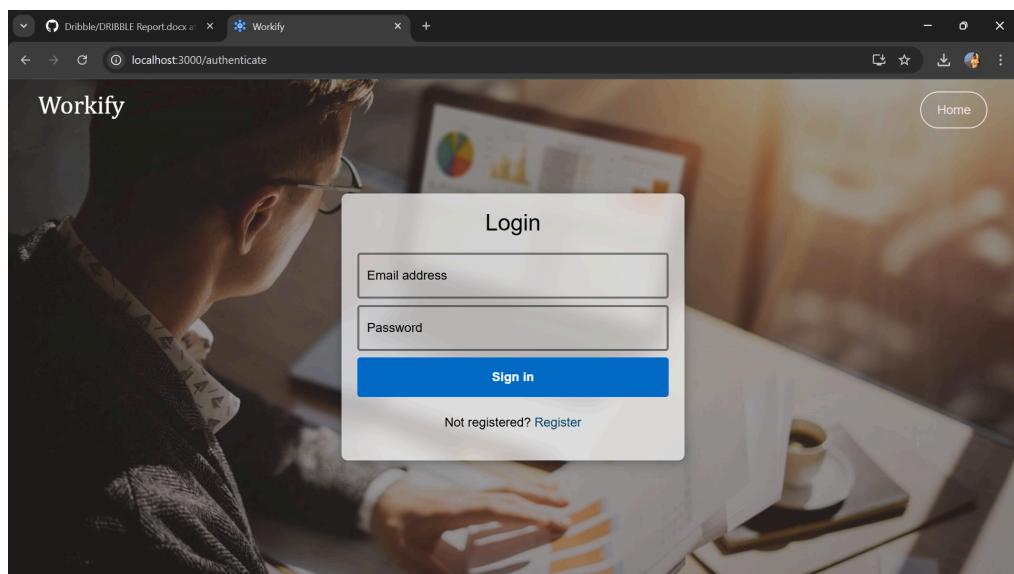
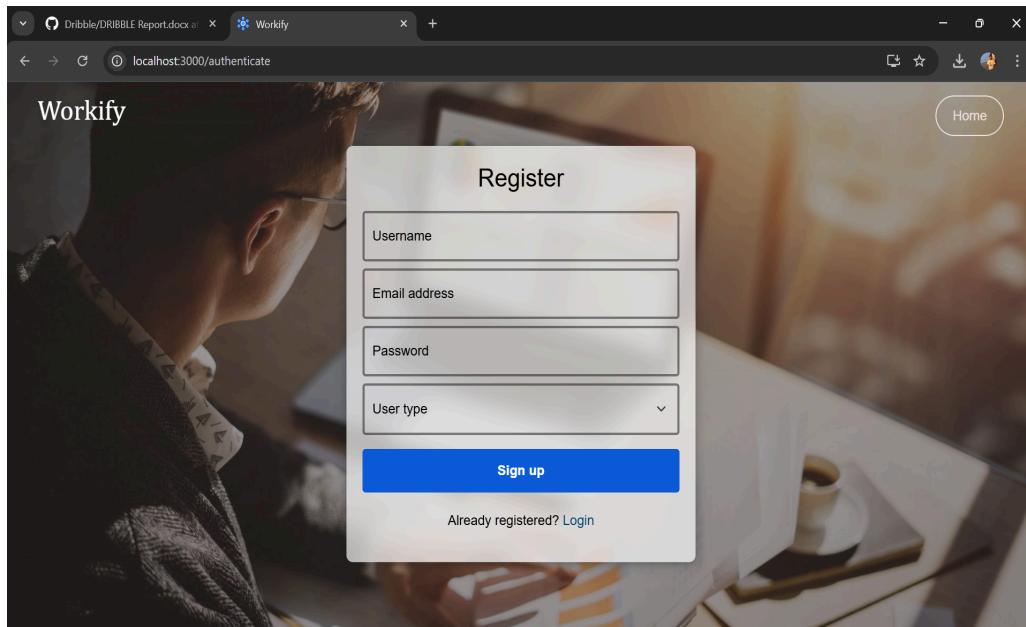
```
server
  > node_modules
  & .env
  JS index.js
  {} package-lock.json
  {} package.json
  JS Schema.js
  JS SocketHandler.js
```

Running the Application

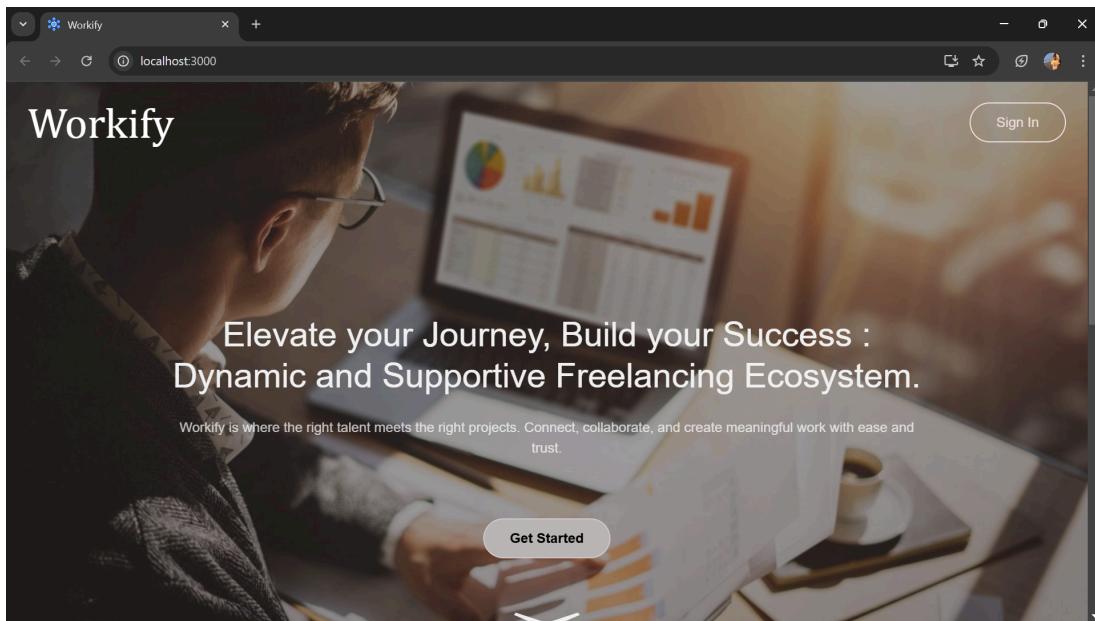
- **Frontend:** “npm start” in the client directory.

- **Backend:** “npm start” in the server directory.

Authentication



USER INTERFACE



A screenshot of the 'About Us' page from the Workify website. The title bar shows 'Workify' and the address bar shows 'localhost:3000'. The main visual is a photograph of a person working on a laptop. The heading 'About Us' is centered above the text. The text describes the platform's mission: 'Welcome to Workify, the ultimate platform where talent and opportunity come together! Workify connects clients with skilled freelancers, creating a thriving community for seamless project collaboration. Clients can post their project needs, while freelancers from various fields bid to bring their expertise to the table. With an efficient process that promotes quality and timely completion, Workify is dedicated to helping both clients and freelancers reach their full potential through successful partnerships.' Below this, two sections are titled 'Role of Clients' and 'Role of Freelancers', each with a brief description of their respective roles on the platform.

SCREENSHOTS

Client's Dashboard

The screenshot shows a web browser window titled 'Workify' at the URL 'localhost:3000/client'. The main content area is titled 'My Projects' and displays three project cards:

- Online Pharmacy Website Development** (Mon Nov 25 2024 22:08:26)
Budget - ₹ 2000
A comprehensive platform for ordering medicines online, accessing health information, and managing prescriptions securely.
Status - Assigned
- Bike Rental App Development** (Mon Nov 25 2024 14:17:13)
Budget - ₹ 20000
A mobile app that allows users to rent bikes, view available locations, make payments, and manage bookings.
Status - Completed
- Cross-Platform E-commerce Mobile App Development** (Mon Nov 25 2024 13:05:42)
Budget - ₹ 25000
Develop a scalable e-commerce mobile solution for businesses to reach a wider customer base through iOS and Android platforms.

Add Projects

The screenshot shows a web browser window titled 'Workify' at the URL 'localhost:3000/new-project'. The main content area is titled 'Post new project' and contains a form with the following fields:

- Project title
- Description
- Budget (in ₹)
- Required skills (separate each with ,.)

At the bottom of the form is a large red-bordered 'Submit' button.

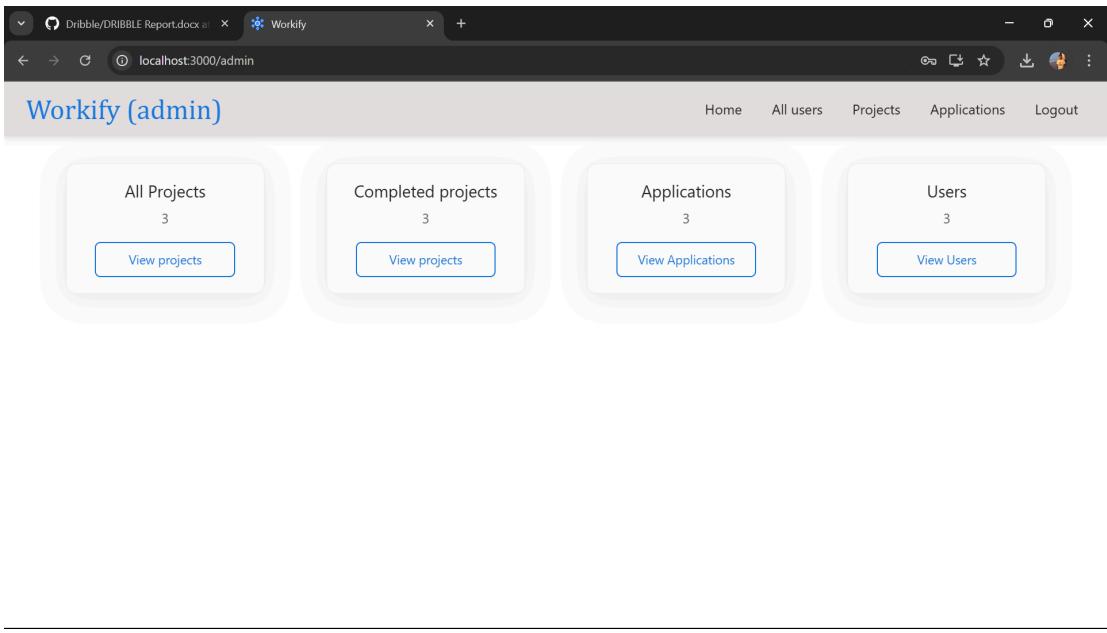
Freelancer Dashboard

The screenshot shows a browser window titled "Workify" at the URL "localhost:3000/freelancer". The dashboard features a top navigation bar with links for "Dashboard", "All Projects", "My Projects", "Applications", and "Logout". Below the navigation is a row of four cards: "Current projects" (1, "View projects"), "Completed projects" (4, "View projects"), "Applications" (3, "View Applications"), and "Funds" (Available: ₹ 60200). A large central panel contains sections for "My Skills" (No skills available), "Description" (please add your description), and an "Update" button.

Freelancer's Project Search

The screenshot shows a browser window titled "Workify" at the URL "localhost:3000/all-projects". The interface includes a top navigation bar with links for "Dashboard", "All Projects", "My Projects", "Applications", and "Logout". On the left, a "Filters" sidebar lists various skills with checkboxes: Javascript, Angular, Figma, Stripe, Kotlin, Express.js, Mongo DB, Sketch, Flutter, SQL, Node.js, and Adobe XD. The main area is titled "All projects" and displays two project listings: "Online Pharmacy Website Development" (Budget ₹ 2000, Mon Nov 25 2024 22:08:26) and "Bike Rental App Development" (Budget ₹ 20000, Mon Nov 25 2024 14:17:13). Each listing includes a brief description, a list of required skills (e.g., Javascript, Angular, Figma, Stripe for the first project), and bidding information ("1 bids ₹ 2000 (avg bid)" for the first project).

Admin Dashboard



Future Enhancements

- User Dashboard Enhancements: Creating personalized dashboards for clients and freelancers, showing key metrics such as ongoing projects, earnings, and pending actions.
- Multi-Language Support: Adding multi-language capabilities to make the platform accessible to a global user base, improving user experience and engagement.
- Automated Invoicing System: Implementing an automated invoicing feature for freelancers to generate and send invoices to clients after project completion.

Demonstration Video Link:

https://drive.google.com/file/d/1a2_FcdDLUM-qiGoIcMs6kTrlVEf_8OXw/view?usp=sharing

