PROJECT SYNOPSIS REPORT

ON

Freelance Skill Exchange (FxE)

SUBMITTED

TO

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FOR

Full Stack Engineering

Submitted By:
Kartik Sharma
(2210991759)

Semester: 6th
Session: 2022-2026

ACKNOWLEDGEMENT

I am deeply thankful to my project guide **Ms. Preenu Mittan** for her invaluable support, insightful suggestions, and encouragement during the development of this project. Her guidance has been instrumental in shaping this work.

Lastly, I extend my appreciation to the **Computer Science & Engineering Department** and my university for providing the necessary resources and a conducive environment for successfully completing this project.

DECLARATION

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person, nor material that has been accepted for the award of any other degree or diploma at any university or institution of higher learning, except where due acknowledgment has been made in the text.

Date : 06-03-2025

Place : Chitkara University

Kartik Sharma

2210991759

Signature

CHITKARA UNIVERSITY

TABLE OF CONTENT

INTRODUCTION

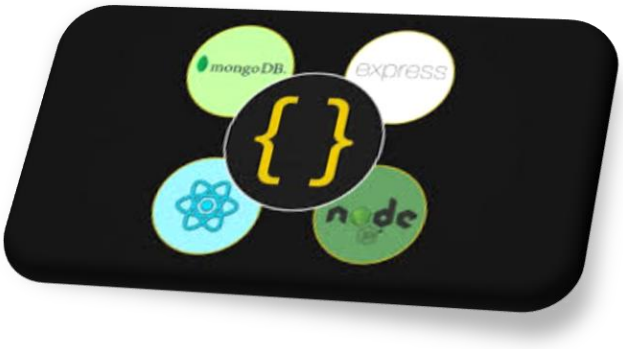
Developers worldwide are continuously striving to enhance user experience and optimize developer workflows for building applications efficiently. Modern web development relies on technology stacks that provide pre-built frameworks and tools to expedite project delivery and streamline development.

The MERN stack is a widely adopted JavaScript-based stack that offers a powerful framework for building full-stack applications.

It consists of MongoDB, Express.js, React.js, and Node.js, each serving a specific role in the development process.

- MongoDB is a NoSQL database that enables flexible and scalable data storage.
- Express.js simplifies backend development by providing a lightweight web framework for Node.js.
- React.js is a powerful frontend library for building dynamic user interfaces.
- Node.js acts as the runtime environment, allowing JavaScript to be used on the server side.By leveraging MERN, developers can build scalable and high-performance web applications while maintaining a single programming language across the entire stack.

This enhances productivity, reduces development overhead, and ensures seamless integration between frontend and backend components.



1.1 Problem Statement

Freelancers often face challenges in finding the right opportunities and collaborators. Existing platforms charge high commissions, lack a streamlined authentication process, and fail to provide an engaging interface for showcasing skills. Our **Freelance Skill Exchange Platform** aims to bridge this gap by creating a secure and interactive ecosystem where professionals can exchange skills without intermediaries.

1.2 Innovative Idea of Project

The platform will integrate:

- **Project showcase via video reels** to help freelancers present their work.
- **OTP-based authentication** for secure access.
- **Real-time chat functionality** to enhance communication.
- **Profile verification system** for ensuring credibility.
- **Skill matching** to connect users with relevant collaborations.
- **A community-driven model** with minimal transaction costs.

2. Scope

The project is designed for freelancers, students, and professionals seeking collaboration. It will include AI-based recommendations, a structured onboarding process, and an interactive dashboard for tracking projects. Users will be able to exchange skills, showcase their work, and collaborate efficiently.

3. What is Mongo DB ?

MongoDB is a **NoSQL database** used for managing large-scale data with high flexibility. It stores information in **JSON-like documents** instead of traditional relational tables, making it highly scalable.

3.1 Key Components of Mongo DB

- **_id** – A 24-digit unique identifier required in every MongoDB document.
- **Collection** – A group of MongoDB documents, similar to an RDBMS table.
- **Document** – A set of key-value pairs with dynamic schema, allowing flexibility in data storage.
- **Database** – A container for collections, each database gets its own set of files.
- **Field** – A name-value pair in a document, analogous to columns in relational databases.
- **Indexing** – Improves query performance and retrieval speed.
- **Replication** – Ensures high availability by duplicating data across multiple servers.
- **Sharding** – Enables horizontal scaling by distributing data across multiple machines.

4. What is Express JS ?

- Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications. It is an open source framework developed and maintained by the Node.js foundation.

- Express provides us the tools that are required to build our app, be it single-page, multi-page or hybrid web applications. It is flexible as there are numerous modules available on npm(Node Package Manager), which can be directly plugged into Express.

- Unlike its competitors like Rails and Django, which have an opinionated way of building applications, Express has no "best way" to do something. It is very flexible and pluggable. •Pug (earlier known as Jade) is a terse language for writing
- HTML templates. It produces HTML, supports dynamic code and code reusability (DRY). It is one of the most popular template languages used with Express.

- Express can be thought of as a layer built on the top of the Node.js that helps manage a server and routes. It allows users to setup middleware to respond to HTTP Requests and defines a routing table which is used to perform different actions based on HTTP method and URL.

- Express allows to dynamically render HTML Pages based on passing arguments to templates. •Express is asynchronous and single threaded and performs I/O operations quickly.

5. What is React ?

- ReactJS is a declarative, efficient, and flexible JavaScript library for building reusable UI components. It is an open-source, component-based front-end library which is responsible only for the view layer of the application. It was initially developed and maintained by Facebook and later used in its products like WhatsApp & Instagram.

- A ReactJS application is made up of multiple components, each component responsible for outputting a small, reusable piece of HTML code. The components are the heart of all React applications. These Components can be nested with other components to allow complex applications to be built of simple building blocks. ReactJS uses virtual DOM based mechanism to fill data in HTML DOM. The virtual DOM works fast as it only changes individual DOM elements instead of reloading complete DOM every time.

- Instead of using regular JavaScript, React codes are written in something called JSX (JavaScript Syntax Extension). JSX is basically a syntax extension of regular JavaScript and is used to create React elements. These elements are then rendered to the React DOM. JSX is faster than normal JavaScript as it performs optimizations while translating to regular JavaScript.

6. What is Node JS ?

- Node.js is a very powerful JavaScript-based platform built on Google Chrome's JavaScript V8 Engine. It is used to develop I/O intensive web applications like video streaming sites, single- page applications, and other web applications. Node.js is open source, completely free, and used by thousands of developers around the world.

- Node.js is a server-side platform built on Google Chrome's JavaScript Engine (V8 Engine). Node.js was developed by Ryan Dahl in 2009.

- Node.js applications are written in JavaScript and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.

- Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

6.1 Features of Node JS ?

- Extremely fast: Node.js is built on Google Chrome's V8 JavaScript Engine, so its library is very fast in code execution.

- I/O is Asynchronous and Event Driven: All APIs of Node.js library are asynchronous i.e. non-blocking. So, a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call. It is also a reason that it is very fast.

- Single threaded: Node.js follows a single threaded model with event looping.

- Highly Scalable: Node.js is highly scalable because event mechanism helps the server to respond in a non-blocking way.

- No buffering: Node.js cuts down the overall processing time while uploading audio and video files. Node.js applications never buffer any data. These applications simply output the data in chunks.

- Open source: Node.js has an open source community which has produced many excellent modules to add additional capabilities to Node.js application.

7. Product Overview

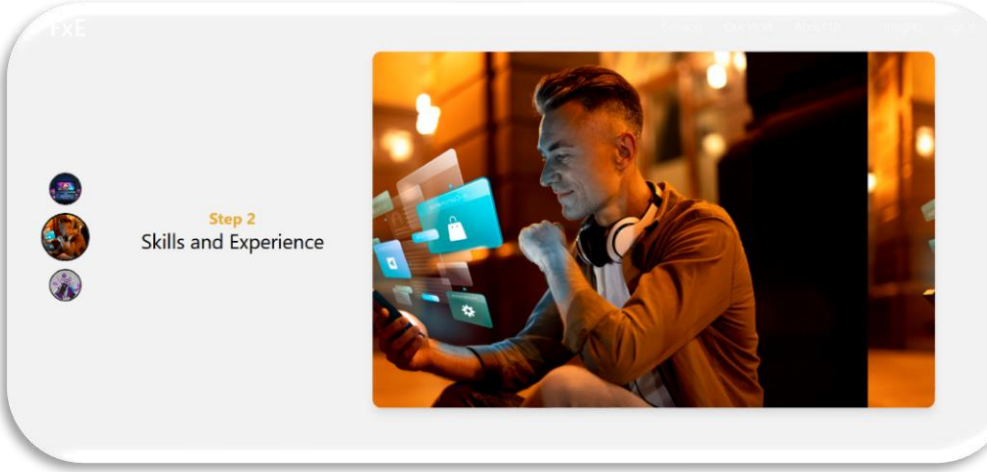
7.1 Interface

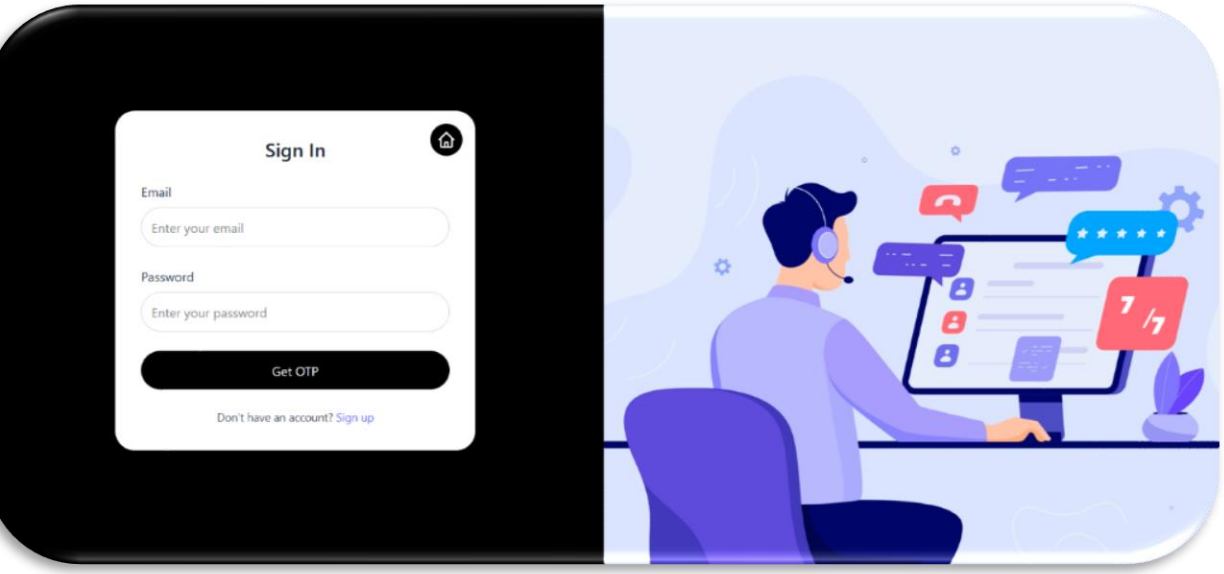
The Interface is designed using React JS & Tailwind CSS, ensuring a seamless experience. Additionally,Framer Motion and GSAP is used to enhance animations and transitions, making the platform making the platform visually appealing and interactive

7.2 Features of Application

- Skill exchange mechanism.
- Video-based project showcase.
- Real-time messaging.
- Profile verification.
- Animated UI using Framer Motion and GSAP

8. Project GUI Screenshots





9. Partial Schema Structure

User Schema

```
const mongoose = require('mongoose');

const UserSchema = new mongoose.Schema({
  email: {
    type: String,
    required: true,
    unique: true,
    lowercase: true,
  },
  password: {
    type: String,
    required: true,
  },
}, { timestamps: true });

module.exports = mongoose.model('User', UserSchema);
```

Profile Schema

```
const mongoose = require('mongoose');

const profileSchema = new mongoose.Schema({
  userId: { type: mongoose.Schema.Types.ObjectId, ref: 'User', required: true, unique: true
}, // Reference to the User
  skills: [{ type: String }],
  expertiseLevel: { type: String, enum: ['Beginner', 'Intermediate', 'Expert'] },
  portfolio: [{ type: String }],
  bio: { type: String },
  verified: { type: Boolean, default: false },
});

module.exports = mongoose.model('Profile', profileSchema);
```

Projects Schema

```
// models/projectSchema.js
const mongoose = require('mongoose');

const projectSchema = new mongoose.Schema({
  userId: { type: mongoose.Schema.Types.ObjectId, ref: 'User', required: true }, //
Reference to the user who uploaded the project
  title: { type: String, required: true },
  description: { type: String, required: true },
  screenshots: [{ type: String }],
  liveLink: { type: String },
  videoLink: { type: String },
  createdAt: { type: Date, default: Date.now },
});

module.exports = mongoose.model('Project', projectSchema);
```

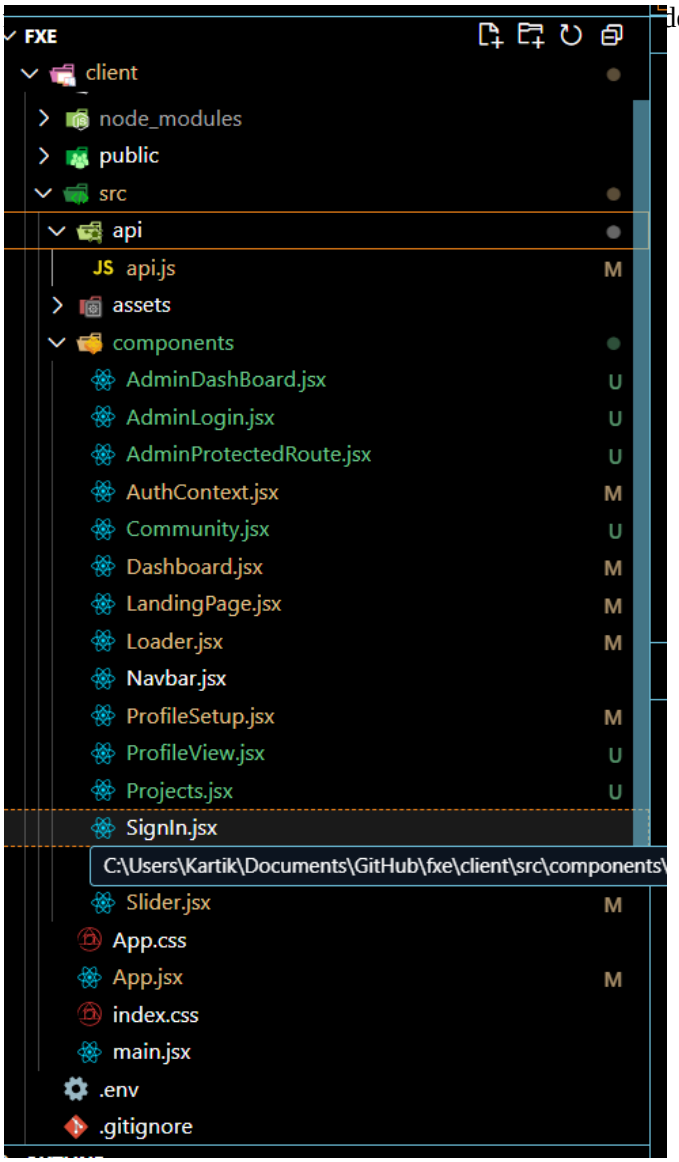
Message Model Schema

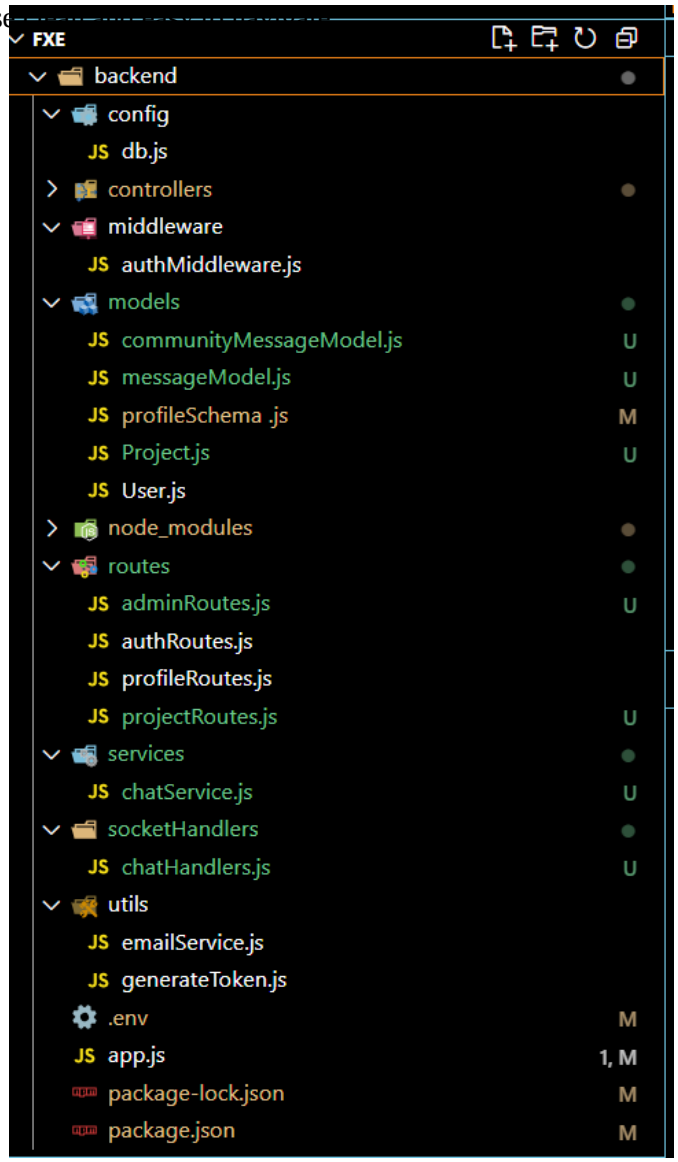
```
const mongoose = require('mongoose');

const messageSchema = new mongoose.Schema({
  id: { type: String, required: true, unique: true },
  sender: { type: String, required: true },
  receiver: { type: String, required: true },
  text: { type: String, required: true },
  timestamp: { type: Date, default: Date.now },
  read: { type: Boolean, default: false }
});

const Message = mongoose.model('Message', messageSchema);
module.exports = Message;
```

10. Maintaining the Folder Structure of Client and Server

A well-structured project architecture is crucial for scalability, maintainability, and collaboration. The **Freelance Skill Exchange Platform** follows a **modular folder structure** to separate concerns between the **client (frontend)** and **server (backend)**.

Since the beginning of the project, a **consistent and modular folder structure** has been maintained to ensure **scalability, maintainability, and efficient development**. The structure follows best practices for organizing both the codebase





11. Hardware Requirements

- Minimum **4GB RAM**, recommended **8GB RAM**.
- At least **Intel i5 processor** or equivalent.
- Internet connection for seamless interaction.

12. Software Requirements

12.1 Software Requirements for Developers

- Operating System: **Windows/Linux/MacOS**
- Backend: **Node.js & Express.js**
- Database: **MongoDB**
- Frontend: **React.js & Tailwind CSS**

12.2 Software Requirements for Users (When Deployed and Live)

Any Device with internet connection to visit the deployed website

13. Conclusion

The Freelance Skill Exchange Platform aims to revolutionize freelancer collaboration by providing a **secure, efficient, and engaging** way to exchange skills.

14. Future Plans

- Implementing Skill Matchmaking System
- Location Based Profile + Matchmaking
- Escrow System for successful deals.

15. References

MongoDB: https://www.mongodb.com/docs/

ExpressJS: https://expressjs.com/

Npm: https://docs.npmjs.com/

NodeJS: https://nodejs.org/docs/latest/api/

WebSocket: https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API, https://socket.io/docs/v4/

REST API: https://docs.github.com/en/rest

Framer Motion : https://motion.dev/

GSAP : https://gsap.com/docs/v3/