PROJECT OVERVIEW

Millions of skilled individuals treat their talents as mere hobbies, missing out on potential income. A software engineer with an eye for design, a professor with poetic brilliance, or a musician with a weekend gig—many have skills that could be monetized, but they lack the right platform to showcase their abilities. On the other hand, businesses struggle to find reliable freelancers due to vague reviews, inefficient searches, and delayed payments.

- **The Problem**

  - ❖ Freelancers rely on ratings that may not accurately reflect their true capabilities.
  - ❖ Job providers struggle to find quality talent quickly and efficiently.
  - ❖ Payment security and delays discourage freelancers from taking up more work.

  ➔ **The WeSkill Solution**
  WeSkill is an AI-powered freelancing platform that revolutionizes how freelancers and job providers connect. With **AI-driven sentiment analysis**, it evaluates reviews to offer **a transparent and reliable rating system**, ensuring that job providers can **make informed hiring decisions**.

Our **smart recommendation engine** personalizes freelancer discovery using a tailored **questionnaire-based matching system**, reducing the hassle of endless searches. Additionally, every freelancer gets a **QR code linked to their UPI ID**, ensuring **instant and secure payments**—eliminating delays and uncertainties.

★ **Key Features & Benefit**

- ❖ **AI-Powered Sentiment Analysis** – Ensures credibility with unbiased insights on freelancer quality.
- ❖ **Smart Matching Algorithm** – Connects job providers with ideal freelancers based on skill, industry, and needs.
- ❖ **Secure & Instant Payments** – QR code-based transactions eliminate payment delays and ensure fair compensation.
- ❖ **Profile Badges & Recognition** – Enhances freelancer visibility and trust with verified achievements

Dependencies:

- For AI
  - **Python (python version : 3.10.11) Libraries**
  - **Pandas (version:2.2.3)** – Data handling and processing (CSV files).
  - **Joblib(version:1.4.2)** – Loading and saving the trained model
  - **Spacy(version:3.8.4)** – Natural language processing (NLP)
  - **scikit-learn(version:1.6.1)** – Machine learning utilities:
    - **TfidfVectorizer** – Convert text into numerical features
    - **MultinomialNB** – Sentiment classification using Naïve Bayes.
    - **make_pipeline** – Creating ML pipelines.
  - **Machine Learning Model** - Naïve Bayes Classifier (MultinomialNB) – Used for sentiment analysis.
  - **NLP Model** - SpaCy Pretrained Model (en_core_web_sm) – Text preprocessing.
  - **Dataset** - CSV File (DataSet.csv) – Contains sentiment analysis training data.
  - **Trained Model** - Pretrained Sentiment Model **(sentiment_model.pkl**) – Predicts sentiment from text.

- For Backend
  - express – A fast and minimal web framework for handling routes and API requests.
  - mongoose – An ODM (Object Data Modeling) library that simplifies working with MongoDB.
  - dotenv – Loads environment variables from a .env file to manage configurations securely.
  - jsonwebtoken (JWT) – Handles user authentication and authorization through secure tokens.
  - bcryptjs – Hashes passwords before storing them in the database for security.
  - cors – Enables Cross-Origin Resource Sharing (CORS) for smooth frontend-backend communication.
  - nodemon – Restarts the server automatically on file changes, useful for development.
  - razorpay – Integrates Razorpay for handling online payments.
  - qrcode – Generates QR codes as images or data URLs for authentication or other purposes.
- For Frontend
  - react-router-dom – Manages routing and navigation within the React application
  - axios – Handles API requests between the frontend and backend.
  - @/components/ui/card – UI component used for profile cards and other visual elements.
  - bootstrap – Provides responsive styling (only used for code-related UI components).

- qrcode.react – Generates QR codes in React components for display.
- react-qr-reader – Allows users to scan QR codes using their device camera.

Setup Instructions:

❖ **FOR AI :**

- **python -m venv myenv**
- **pip install pandas**
- **pip install joblib**
- **pip install spacy**
- **Pip install scikit-learn**
- **python -m spacy download en_core_web_sm**

❖ **FOR FRONTEND:**

Prerequisites

- Ensure you have the following installed before proceeding:

  - [Node.js](https://nodejs.org/) (Latest LTS version recommended)

  - [MongoDB](https://www.mongodb.com/try/download/community) (For database management)

  - [Git](https://git-scm.com/downloads)

- **Clone the Repository (write all this in terminal of vs code)**

  bash

  git clone https://github.com/Love-M-365/weSkill

  cd weskill

- **Backend Setup**

  1. Navigate to the backend folder:

  bash

cd backend

2. Install dependencies:

bash

npm install

3. Start the backend server:

bash

npm run dev

- **Frontend Setup**

1. Navigate to the frontend folder:

bash

cd ../frontend

2. Install dependencies:

bash

npm install

3. Start the frontend:

bash

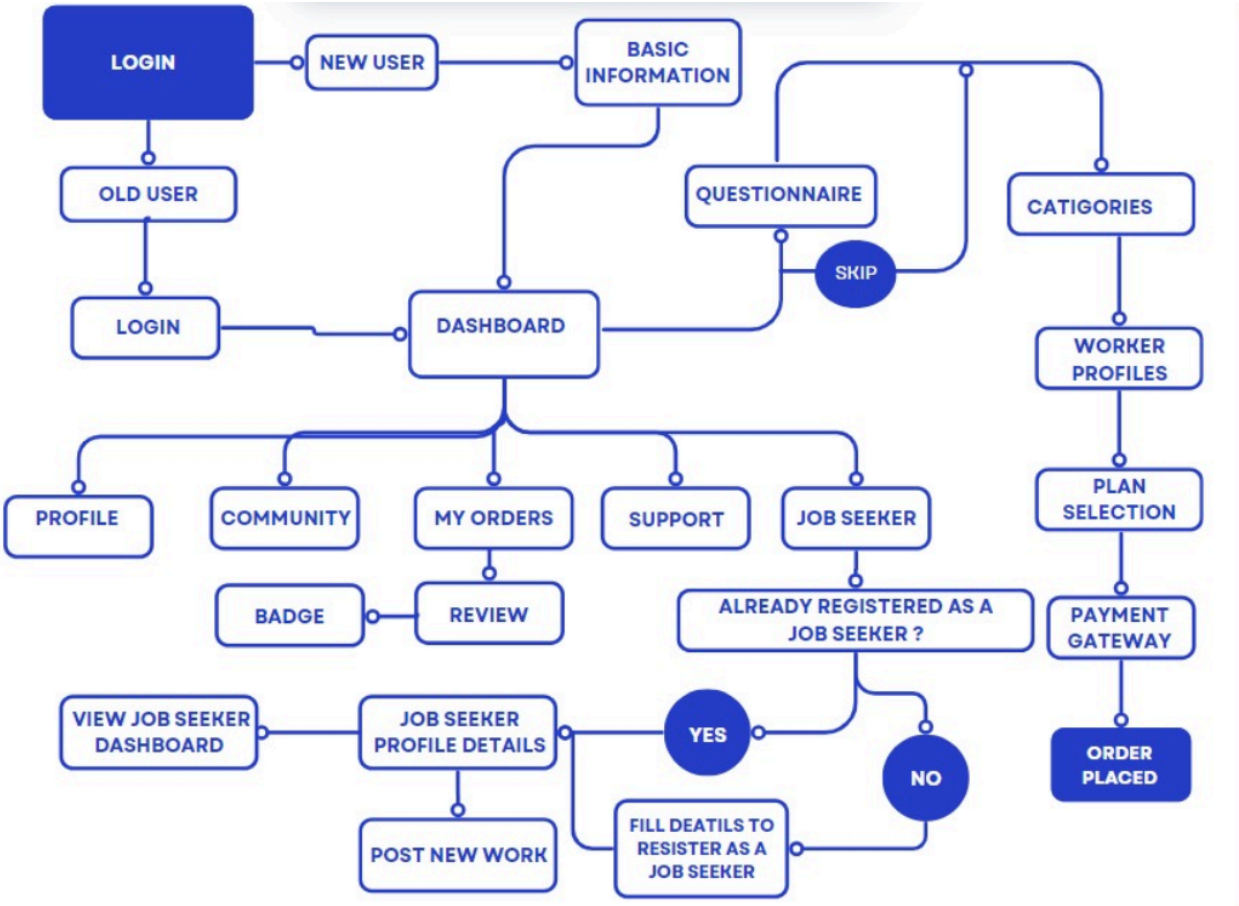
npm start(The frontend will run on http://localhost:3000)

- **Running the Application**

Once both frontend and backend are running:

- Open http://localhost:3000 in your browser.

- The WeSkill dashboard should be visible.



FUTURE PLANS:

- Advanced AI-Based Matching – Use machine learning to predict the best freelancers for a job based on skills, experience, and past work.
- Hold payments securely until work is approved, ensuring **trust** between freelancers and clients.
- Enable **cross-border transactions** for global clients and freelancers.
- Notify freelancers about **high-paying gigs** based on their skills.
- Create a **dedicated mobile app** for easier navigation, notifications, and seamless transactions.
- A dynamic **leaderboard** ranking freelancers based on **past work, ratings, earnings, response time, and reliability**.