Project Documentation

Full Stack Development with Flask & TensorFlow

1. Introduction

Project Title: FreelanceFinder: Discovering Opportunities, Unlocking Potential

Team Leader:
Katragadda Uday Kiran

Team Members:

- Kamineni Sai Harika
- Kalpana Korni
- Kari Harshitha

2. Project Overview

Purpose:

FreelanceFinder is a web-based platform designed to match freelancers with suitable job opportu nities based on their skills and preferences. It leverages machine learning models for intellig ent job recommendations and provides a clean, intuitive interface for seamless interaction.

Features:

- Real-time job and freelancer matching
- Flask-based responsive web UI
- Skill-based filtering using machine learning
- Dashboard for both freelancers and recruiters

3. Architecture

Frontend:
HTML and Bootstrap templates served by Flask (home.html, profile.html, dashboard.html)
Backend:
Python Flask handles routing, authentication, and interaction with the ML model in app.py
Model:
Recommendation engine trained on freelance job datasets to map skills with job postings

4. Setup Instructions

Prerequisites:

- Python 3.8+
- Flask
- Scikit-learn / TensorFlow
- Pandas, NumPy

Installation Steps:

```
git clone https://github.com/YOUR_USERNAME/FreelanceFinder.git
cd FreelanceFinder pip install -r requirements.txt python app.py
```

5. Folder Structure

```
FreelanceFinder/
├── app.py
├── model.pkl
├── requirements.txt
├── templates/
│   ├── home.html
│   ├── profile.html
│   └── dashboard.html
└── static/
    └── (CSS, JS, uploaded files)
```

6. Running The Application

Frontend: Flask serves the UI forms and dashboards

Backend: Launch with: python app.py

Access: Open http://127.0.0.1:5000/ in a browser

7. API Documentation

Endpoint: /match

Method: POST

Input: JSON payload with freelancer profile or job post

Output: JSON with ranked matches or HTML rendered view

8. Authentication

Current Version: Basic session-based login system for both freelancers and employers

Future Plan: OAuth and token-based API access

9. User Interface

Pages:

- home.html – Landing & search
- profile.html – Freelancer registration/profile
- dashboard.html – Job and talent dashboard

10. Testing

Unit and integration testing with sample users and job datasets Cross-browser UI testing (Chrome, Firefox, Edge)

11. Screenshots or Demo

- Landing Page
- Job Match Result Page
- Freelancer Dashboard

12. Known Issues

- Cold-start problem for new users (no past data)
- Limited dataset support in initial version

13. Future Enhancements

- Integrate advanced AI/ML for better job predictions
- Add mobile app version
- Enable chat and contract system between freelancers and recruiters
- Expand skill matching with external APIs (LinkedIn, GitHub)