Introduction

FreelanceFusion is a blockchain-powered freelance platform that integrates smart contracts and automated dispute resolution. This project includes payment integration to ensure seamless financial transactions. Developed as part of a BSc (Hons) Computer Science and Software Engineering undergraduate thesis at the University of Bedfordshire .

FreelanceFusion Installation & Deployment Guide

Version: 1.0

Overview

Description:

Blockchain-Integrated Freelance Platform

Project Structure:

```
/freelancefusionbackendserver
/freelancefusionfrontend
/freelancefusionsmartcontract
```

Table of Contents

System Requirements

Minimum Specifications

- **OS:** Ubuntu 20.04 LTS / macOS Monterey 12.5+ / Windows 10+ (WSL2 recommended)
- **Memory:** 8GB RAM (16GB recommended for blockchain)
- **Storage:** 20GB free space (SSD recommended)

Required Software

Software	Version	Verify Command	Installation Guide
Node.js	16.x+	`node -v`	https://nodejs.org (https://nodejs.org)
Python	3.9+	`python3 --version`	https://github.com/pyenv/pyenv (https://github.com/pyenv/pyenv)
PostgreSQL	13+	`psql --version`	`sudo apt install postgresql postgresql-contrib`
Ganache CLI	7.6.0+	`ganache --version`	`npm install -g ganache`
Truffle Suite	5.6.0+	`truffle version`	`npm install -g truffle`
npm	8.x+	`npm -v`	Bundled with Node.js

Repository Setup

Step 1: Clone the Repository and Verify Structure

```
git clone https://github.com/ramesh-sah/freelancefusion.git
cd freelancefusion

# Verify structure
ls -l
# freelancefusionbackend/
# freelancefusionfrontend/
# freelancefusionsmartcontract/
# README.md
```

Step 2: Install Dependencies

```
cd freelancefusionbackend && pip install -r requirements.txt
cd ../freelancefusionfrontend && npm install
cd ../freelancefusionsmartcontract && npm install
```

Database Configuration

PostgreSQL Setup

1. **Login:**

```
sudo -u postgres psql
```

2. **Create DB and User:**

```
CREATE DATABASE freelancefusion;
CREATE USER fusion_admin WITH PASSWORD 'securepassword123';
ALTER ROLE fusion_admin SET client_encoding TO 'utf8';
ALTER ROLE fusion_admin SET default_transaction_isolation TO 'read committed';
GRANT ALL PRIVILEGES ON DATABASE freelancefusion TO fusion_admin;
\q
```

Environment Configuration

```
cd backend
cp .env.example .env
```

Update `.env` with:

```
DATABASE_URL=postgresql://fusion_admin:securepassword123@localhost:5432/freelancefusion
SECRET_KEY=your_django_secret_key_here
```

Apply Migrations

```
python manage.py makemigrations
python manage.py migrate
```

Backend Service

Development Mode

```
cd backend
python manage.py runserver 0.0.0.0:8000
```

Verify:

```
curl http://localhost:8000/api/healthcheck
# Expected: {"status": "ok"}
```

Production Configuration

```
pip install gunicorn
gunicorn --workers 4 --bind 0.0.0.0:8000 core.wsgi:application
```

Blockchain Network

Local Blockchain Setup

```
ganache --chain.chainId 1337 --wallet.totalAccounts 10 --wallet.defaultBalance 100 --server.port 8545
```

Deploy Contracts:

```
cd smartcontract
truffle migrate --network development
```

Verify Deployment:

```
truffle console --network development
> const contract = await MyContract.deployed()
> console.log(contract.address)
```

Contract Integration

Update `frontend/src/config/contracts.js`:

```
export const CONTRACT_ADDRESS = '0x123...'; // Replace with actual address
export const CONTRACT_ABI = [ /* ABI from build/contracts/MyContract.json */ ];
```

Frontend Application

Development Mode

```
cd frontend
cp .env.example .env.local
```

Update `.env.local`:

```
REACT_APP_API_URL=http://localhost:8000
REACT_APP_BLOCKCHAIN_RPC=http://localhost:8545
```

Start App:

```
npm start
```

Production Build

```
npm run build
serve -s build -l 3000
```

Full System Launch

Service Checklist

- PostgreSQL: `sudo systemctl status postgresql`
- Backend: http://localhost:8000
- Blockchain: Ganache on port 8545
- Frontend: http://localhost:3000

Test Workflow

1. Create user account via frontend
2. Post sample project
3. Submit proposal with another account
4. Check Ganache logs for blockchain transaction

Production Deployment

Infrastructure Recommendations

- **Backend:** NGINX + Gunicorn
- **Frontend:** Vercel or Netlify
- **Blockchain:** Infura or Alchemy
- **Database:** AWS RDS or Google Cloud SQL

Security (NGINX Configuration)

```
server {
    listen 80;
    server_name yourdomain.com;

    location /api {
        proxy_pass http://localhost:8000;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $host;
        proxy_redirect off;
    }

    location / {
        root /var/www/frontend/build;
        try_files $uri $uri/ /index.html;
    }
}
```

Troubleshooting

Common Issues

Symptom	Solution
Database connection refused	Check PostgreSQL service
Migration errors	`python manage.py migrate --fake`
Contract deployment failures	Verify Ganache gas limit in `truffle-config`
CORS errors	Add frontend URL to `CORS_ORIGIN_WHITELIST`
Insufficient funds	Use `truffle develop` for test accounts

Log Locations

- **Backend:** `backend/core/logs/django.log`
- **Frontend:** Browser Console, `npm run test:debug`
- **Blockchain:** Ganache terminal

Emergency Reset

```
sudo -u postgres psql -c "DROP DATABASE freelancefusion;"
cd backend && rm -rf migrations/
python manage.py makemigrations
truffle migrate --reset --network development
```

Contact:

 rsah3798@gmail.com (mailto:rsah3798@gmail.com)
