Freelenso – A Freelanceing platform

Django naturally aligns with the **MVC** architecture:

- **Model**: Manages the database schema (users, projects, payments, etc.)
- **View**: Handles user interface logic (project listings, proposals, profiles).
- **Controller**: Manages business logic (user authentication, payments, etc.)

Database Schema Design

USERS

```
CREATE TABLE users (
    id INT PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(50) UNIQUE NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    password VARCHAR(255) NOT NULL,
    role ENUM('client', 'freelancer', 'admin') NOT NULL,
    profile_image VARCHAR(255) DEFAULT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Freelancer Profiles

```
CREATE TABLE freelancer_profiles (
    id INT PRIMARY KEY AUTO_INCREMENT,
    user_id INT NOT NULL,
    skills TEXT NOT NULL,
    portfolio_link VARCHAR(255),
    bio TEXT,
    rating FLOAT DEFAULT 0,
    review_count INT DEFAULT 0,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE
CASCADE
);
```

Client Profiles

```
CREATE TABLE client_profiles (
    id INT PRIMARY KEY AUTO_INCREMENT,
    user_id INT NOT NULL,
    company_name VARCHAR(255),
    bio TEXT,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE
CASCADE
);
```

Projects

```
CREATE TABLE projects (
    id INT PRIMARY KEY AUTO_INCREMENT,
    client_id INT NOT NULL,
    title VARCHAR(255) NOT NULL,
    description TEXT NOT NULL,
    budget DECIMAL(10, 2) NOT NULL CHECK (budget > 0),
    deadline DATE NOT NULL,
    status ENUM('open', 'in_progress', 'completed') DEFAULT 'open',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (client_id) REFERENCES users(id) ON DELETE CASCADE
);
```

Proposals

```
CREATE TABLE proposals (
    id INT PRIMARY KEY AUTO_INCREMENT,
    project_id INT NOT NULL,
    freelancer_id INT NOT NULL,
    bid_amount DECIMAL(10, 2) NOT NULL CHECK (bid_amount > 0),
    cover_letter TEXT NOT NULL,
    status ENUM('pending', 'accepted', 'rejected') DEFAULT 'pending',
    submitted_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (project_id) REFERENCES projects(id) ON DELETE CASCADE,
    FOREIGN KEY (freelancer_id) REFERENCES users(id) ON DELETE CASCADE
);
```

Payments (Escrow System)

```
CREATE TABLE payments (
    id INT PRIMARY KEY AUTO_INCREMENT,
    project_id INT NOT NULL,
    client_id INT NOT NULL,
    freelancer_id INT NOT NULL,
    amount DECIMAL(10, 2) NOT NULL CHECK (amount > 0),
    status ENUM('escrow', 'released') DEFAULT 'escrow',
    transaction_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (project_id) REFERENCES projects(id) ON DELETE CASCADE,
    FOREIGN KEY (client_id) REFERENCES users(id) ON DELETE CASCADE,
    FOREIGN KEY (freelancer_id) REFERENCES users(id) ON DELETE CASCADE
);
```

Reviews

```
CREATE TABLE reviews (
    id INT PRIMARY KEY AUTO_INCREMENT,
    project_id INT NOT NULL,
    client_id INT NOT NULL,
    freelancer_id INT NOT NULL,
    rating INT CHECK (rating BETWEEN 1 AND 5),
    review_text TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (project_id) REFERENCES projects(id) ON DELETE CASCADE,
    FOREIGN KEY (client_id) REFERENCES users(id) ON DELETE CASCADE,
    FOREIGN KEY (freelancer_id) REFERENCES users(id) ON DELETE CASCADE
);
```

Notifications

```
CREATE TABLE notifications (
    id INT PRIMARY KEY AUTO_INCREMENT,
    user_id INT NOT NULL,
    message TEXT NOT NULL,
    status ENUM('unread', 'read') DEFAULT 'unread',
    sent_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE
CASCADE
);
```