**Plan**

Databse

-- Users and Authentication

CREATE TABLE users (

    id INTEGER PRIMARY KEY AUTOINCREMENT,

    username VARCHAR(100) UNIQUE NOT NULL,

    email VARCHAR(255) UNIQUE NOT NULL,

    password\_hash VARCHAR(255) NOT NULL,

    full\_name VARCHAR(200),

    date\_of\_birth DATE,

    created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

    last\_login TIMESTAMP,

    is\_active BOOLEAN DEFAULT 1,

    profile\_picture VARCHAR(255),

    preferred\_language VARCHAR(10) DEFAULT 'no'

);

-- User Progress and Statistics

CREATE TABLE user\_progress (

    id INTEGER PRIMARY KEY AUTOINCREMENT,

    user\_id INTEGER NOT NULL,

    total\_quizzes\_taken INTEGER DEFAULT 0,

    total\_questions\_answered INTEGER DEFAULT 0,

    correct\_answers INTEGER DEFAULT 0,

    total\_game\_sessions INTEGER DEFAULT 0,

    total\_game\_score INTEGER DEFAULT 0,

    total\_videos\_watched INTEGER DEFAULT 0,

    videos\_completed INTEGER DEFAULT 0,

    current\_streak\_days INTEGER DEFAULT 0,

    longest\_streak\_days INTEGER DEFAULT 0,

    last\_activity\_date DATE,

    FOREIGN KEY (user\_id) REFERENCES users(id)

);

-- Achievement System

CREATE TABLE achievements (

    id INTEGER PRIMARY KEY AUTOINCREMENT,

    name VARCHAR(100) NOT NULL,

    description TEXT,

    icon\_filename VARCHAR(255),

    points INTEGER DEFAULT 10,

    category VARCHAR(50), -- 'quiz', 'game', 'video', 'general'

    requirement\_type VARCHAR(50), -- 'score', 'streak', 'completion', etc

    requirement\_value INTEGER

);

CREATE TABLE user\_achievements (

    id INTEGER PRIMARY KEY AUTOINCREMENT,

    user\_id INTEGER NOT NULL,

    achievement\_id INTEGER NOT NULL,

    earned\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

    FOREIGN KEY (user\_id) REFERENCES users(id),

    FOREIGN KEY (achievement\_id) REFERENCES achievements(id),

    UNIQUE(user\_id, achievement\_id)

);

-- Enhanced Questions Table

CREATE TABLE questions (

    id INTEGER PRIMARY KEY AUTOINCREMENT,

    question TEXT NOT NULL,

    correct\_option TEXT NOT NULL,

    category TEXT,

    subcategory TEXT,

    difficulty\_level INTEGER DEFAULT 1, -- 1-5

    explanation TEXT, -- Detailed explanation of correct answer

    image\_filename TEXT,

    created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

    updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

    is\_active BOOLEAN DEFAULT 1,

    question\_type VARCHAR(50) DEFAULT 'multiple\_choice' -- 'multiple\_choice', 'true\_false', 'scenario'

);

-- Quiz Sessions

CREATE TABLE quiz\_sessions (

    id INTEGER PRIMARY KEY AUTOINCREMENT,

    user\_id INTEGER NOT NULL,

    quiz\_type VARCHAR(50), -- 'practice', 'timed', 'category', 'mock\_exam'

    category VARCHAR(100),

    total\_questions INTEGER,

    correct\_answers INTEGER,

    time\_spent\_seconds INTEGER,

    score DECIMAL(5,2),

    started\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

    completed\_at TIMESTAMP,

    FOREIGN KEY (user\_id) REFERENCES users(id)

);

CREATE TABLE quiz\_responses (

    id INTEGER PRIMARY KEY AUTOINCREMENT,

    session\_id INTEGER NOT NULL,

    question\_id INTEGER NOT NULL,

    user\_answer TEXT,

    is\_correct BOOLEAN,

    time\_spent\_seconds INTEGER,

    FOREIGN KEY (session\_id) REFERENCES quiz\_sessions(id),

    FOREIGN KEY (question\_id) REFERENCES questions(id)

);

-- Game Components

CREATE TABLE game\_scenarios (

    id INTEGER PRIMARY KEY AUTOINCREMENT,

    name VARCHAR(200) NOT NULL,

    description TEXT,

    scenario\_type VARCHAR(50), -- 'parking', 'highway', 'city', 'weather'

    difficulty\_level INTEGER DEFAULT 1,

    max\_score INTEGER DEFAULT 100,

    time\_limit\_seconds INTEGER,

    config\_json TEXT -- Store game-specific configuration

);

CREATE TABLE game\_sessions (

    id INTEGER PRIMARY KEY AUTOINCREMENT,

    user\_id INTEGER NOT NULL,

    scenario\_id INTEGER NOT NULL,

    score INTEGER DEFAULT 0,

    time\_played\_seconds INTEGER,

    mistakes\_count INTEGER DEFAULT 0,

    completed BOOLEAN DEFAULT 0,

    started\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

    completed\_at TIMESTAMP,

    FOREIGN KEY (user\_id) REFERENCES users(id),

    FOREIGN KEY (scenario\_id) REFERENCES game\_scenarios(id)

);

-- Interactive Videos

CREATE TABLE videos (

    id INTEGER PRIMARY KEY AUTOINCREMENT,

    title VARCHAR(200) NOT NULL,

    description TEXT,

    filename VARCHAR(255),

    youtube\_url VARCHAR(255),

    duration\_seconds INTEGER,

    category VARCHAR(100),

    difficulty\_level INTEGER DEFAULT 1,

    order\_index INTEGER DEFAULT 0,

    thumbnail\_filename VARCHAR(255),

    created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE video\_checkpoints (

    id INTEGER PRIMARY KEY AUTOINCREMENT,

    video\_id INTEGER NOT NULL,

    timestamp\_seconds INTEGER NOT NULL,

    question\_id INTEGER NOT NULL,

    is\_mandatory BOOLEAN DEFAULT 1,

    FOREIGN KEY (video\_id) REFERENCES videos(id),

    FOREIGN KEY (question\_id) REFERENCES questions(id)

);

CREATE TABLE video\_progress (

    id INTEGER PRIMARY KEY AUTOINCREMENT,

    user\_id INTEGER NOT NULL,

    video\_id INTEGER NOT NULL,

    last\_position\_seconds INTEGER DEFAULT 0,

    completed BOOLEAN DEFAULT 0,

    checkpoints\_passed INTEGER DEFAULT 0,

    total\_checkpoints INTEGER DEFAULT 0,

    started\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

    completed\_at TIMESTAMP,

    FOREIGN KEY (user\_id) REFERENCES users(id),

    FOREIGN KEY (video\_id) REFERENCES videos(id)

);

-- Learning Paths

CREATE TABLE learning\_paths (

    id INTEGER PRIMARY KEY AUTOINCREMENT,

    name VARCHAR(200) NOT NULL,

    description TEXT,

    estimated\_hours INTEGER,

    difficulty\_level INTEGER DEFAULT 1,

    icon\_filename VARCHAR(255),

    is\_recommended BOOLEAN DEFAULT 0

);

CREATE TABLE learning\_path\_items (

    id INTEGER PRIMARY KEY AUTOINCREMENT,

    path\_id INTEGER NOT NULL,

    item\_type VARCHAR(50), -- 'quiz', 'video', 'game'

    item\_id INTEGER,

    order\_index INTEGER,

    is\_mandatory BOOLEAN DEFAULT 1,

    FOREIGN KEY (path\_id) REFERENCES learning\_paths(id)

);

CREATE TABLE user\_learning\_paths (

    id INTEGER PRIMARY KEY AUTOINCREMENT,

    user\_id INTEGER NOT NULL,

    path\_id INTEGER NOT NULL,

    progress\_percentage DECIMAL(5,2) DEFAULT 0,

    started\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

    completed\_at TIMESTAMP,

    FOREIGN KEY (user\_id) REFERENCES users(id),

    FOREIGN KEY (path\_id) REFERENCES learning\_paths(id)

);

-- Leaderboards

CREATE TABLE leaderboard\_entries (

    id INTEGER PRIMARY KEY AUTOINCREMENT,

    user\_id INTEGER NOT NULL,

    leaderboard\_type VARCHAR(50), -- 'weekly', 'monthly', 'all\_time'

    category VARCHAR(50), -- 'quiz', 'game', 'overall'

    score INTEGER,

    rank INTEGER,

    period\_start DATE,

    period\_end DATE,

    created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

    FOREIGN KEY (user\_id) REFERENCES users(id)

);

-- Feedback and Reports

CREATE TABLE user\_feedback (

    id INTEGER PRIMARY KEY AUTOINCREMENT,

    user\_id INTEGER,

    feedback\_type VARCHAR(50), -- 'bug', 'suggestion', 'content\_error'

    subject VARCHAR(200),

    message TEXT,

    status VARCHAR(50) DEFAULT 'pending', -- 'pending', 'reviewed', 'resolved'

    created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

    FOREIGN KEY (user\_id) REFERENCES users(id)

);

driving-theory-app/

├── app.py                      # Main application file

├── config.py                   # Configuration settings

├── requirements.txt            # Python dependencies

├── run.py                     # Application runner

├── .env                       # Environment variables

├── .gitignore

├── README.md

│

├── app/

│   ├── \_\_init\_\_.py           # Initialize Flask app

│   ├── models.py             # SQLAlchemy models

│   ├── forms.py              # WTForms for validation

│   │

│   ├── auth/                 # Authentication blueprint

│   │   ├── \_\_init\_\_.py

│   │   ├── routes.py

│   │   ├── forms.py

│   │   └── utils.py

│   │

│   ├── main/                 # Main pages blueprint

│   │   ├── \_\_init\_\_.py

│   │   ├── routes.py

│   │   └── utils.py

│   │

│   ├── quiz/                 # Quiz functionality

│   │   ├── \_\_init\_\_.py

│   │   ├── routes.py

│   │   ├── utils.py

│   │   └── api.py           # API endpoints for AJAX

│   │

│   ├── game/                 # Game functionality

│   │   ├── \_\_init\_\_.py

│   │   ├── routes.py

│   │   ├── game\_logic.py

│   │   └── api.py

│   │

│   ├── video/                # Video learning

│   │   ├── \_\_init\_\_.py

│   │   ├── routes.py

│   │   ├── utils.py

│   │   └── api.py

│   │

│   ├── admin/                # Admin dashboard

│   │   ├── \_\_init\_\_.py

│   │   ├── routes.py

│   │   ├── forms.py

│   │   └── utils.py

│   │

│   ├── api/                  # RESTful API

│   │   ├── \_\_init\_\_.py

│   │   ├── auth.py

│   │   ├── progress.py

│   │   ├── leaderboard.py

│   │   └── utils.py

│   │

│   └── utils/                # Shared utilities

│       ├── \_\_init\_\_.py

│       ├── decorators.py     # Custom decorators

│       ├── helpers.py        # Helper functions

│       └── validators.py     # Input validation

│

├── static/

│   ├── css/

│   │   ├── main.css         # Main stylesheet

│   │   ├── quiz.css         # Quiz-specific styles

│   │   ├── game.css         # Game-specific styles

│   │   ├── video.css        # Video player styles

│   │   └── admin.css        # Admin panel styles

│   │

│   ├── js/

│   │   ├── main.js          # Main JavaScript

│   │   ├── quiz.js          # Quiz interactions

│   │   ├── game/            # Game scripts

│   │   │   ├── engine.js

│   │   │   ├── scenarios.js

│   │   │   └── controls.js

│   │   ├── video-player.js  # Interactive video player

│   │   └── admin.js         # Admin functionality

│   │

│   ├── images/

│   │   ├── signs/           # Traffic signs

│   │   ├── scenarios/       # Game scenarios

│   │   ├── achievements/    # Achievement icons

│   │   ├── ui/             # UI elements

│   │   └── profiles/        # User avatars

│   │

│   ├── videos/              # Video files

│   └── audio/               # Sound effects

│

├── templates/

│   ├── base.html            # Base template

│   ├── index.html           # Landing page

│   │

│   ├── auth/

│   │   ├── login.html

│   │   ├── register.html

│   │   └── profile.html

│   │

│   ├── quiz/

│   │   ├── quiz\_home.html

│   │   ├── quiz\_session.html

│   │   ├── quiz\_results.html

│   │   └── quiz\_review.html

│   │

│   ├── game/

│   │   ├── game\_home.html

│   │   ├── game\_play.html

│   │   └── game\_results.html

│   │

│   ├── video/

│   │   ├── video\_library.html

│   │   ├── video\_player.html

│   │   └── video\_progress.html

│   │

│   ├── progress/

│   │   ├── dashboard.html

│   │   ├── achievements.html

│   │   └── leaderboard.html

│   │

│   ├── admin/

│   │   ├── dashboard.html

│   │   ├── questions.html

│   │   ├── users.html

│   │   ├── content.html

│   │   └── analytics.html

│   │

│   └── components/         # Reusable components

│       ├── navbar.html

│       ├── footer.html

│       ├── progress\_bar.html

│       └── achievement\_popup.html

│

├── migrations/             # Database migrations

├── tests/                  # Test files

│   ├── test\_quiz.py

│   ├── test\_game.py

│   ├── test\_video.py

│   └── test\_api.py

│

└── scripts/               # Utility scripts

    ├── init\_db.py        # Initialize database

    ├── import\_questions.py

    └── generate\_stats.py

3. Key Features to Implement

User Experience Features:

Gamification System

Points and XP system

Achievement badges

Daily streaks

Leaderboards

Adaptive Learning

Track weak areas

Personalized question recommendations

Difficulty adjustment

Social Features

Friend system

Challenge friends

Share achievements

Progress Tracking

Visual progress charts

Performance analytics

Study time tracking

Technical Features:

Progressive Web App (PWA)

Offline capability

Push notifications

Mobile-responsive

API Design

RESTful endpoints

JWT authentication

Rate limiting

Real-time Features

Live quiz competitions

Real-time leaderboards

WebSocket for game updates

4. Technology Stack Recommendations

Backend:

Flask (current) → Consider FastAPI for better async support

SQLAlchemy for ORM

Redis for caching and sessions

Celery for background tasks

Frontend:

React or Vue.js for interactive UI

Tailwind CSS for modern styling

Chart.js for progress visualization

Phaser.js for game development

Mobile App:

React Native or Flutter for cross-platform

Share API with web version

Infrastructure:

Docker for containerization

PostgreSQL for production database

AWS S3 or similar for media storage

CI/CD pipeline with GitHub Actions

5. Development Phases

Phase 1: Core Foundation (Weeks 1-4)

User authentication system

Enhanced quiz functionality

Basic progress tracking

Responsive design

Phase 2: Gamification (Weeks 5-8)

Achievement system

Points and XP

Leaderboards

User profiles

Phase 3: Interactive Learning (Weeks 9-12)

Video player with checkpoints

Basic driving game

Learning paths

Performance analytics

Phase 4: Mobile & Polish (Weeks 13-16)

PWA implementation

Mobile app development

Performance optimization

Beta testing

6. Security Considerations

Authentication

Implement proper password hashing (bcrypt)

Session management

CSRF protection

Data Protection

Input validation

SQL injection prevention

XSS protection

GDPR Compliance

Privacy policy

Data export functionality

Account deletion

7. Monetization Options

Freemium Model

Basic features free

Premium for advanced analytics

No ads in premium

Subscription Tiers

Basic: Limited daily questions

Premium: Unlimited + videos

Pro: All features + offline mode

One-time Purchase

Full access

Lifetime updates