Plan

The core purpose of this project is to create a modern, all-in-one platform for anyone preparing for their driver's theory test. The idea is to combine:

- **Interactive quizzes** that cover every category of road signs and traffic rules, complete with images and explanations
- Rich multimedia learning through videos and simulated driving scenarios (e.g. roundabouts, intersections, parking)
- **Personalized progress tracking** so each user can see where they're strong or need more practice
- **Gamification elements** like achievements, streaks and leaderboards to keep motivation high

Ultimately, it's about giving learners a single, engaging environment—on web or mobile—where they can learn, practice, and master everything they need to know to pass their driver's exam.

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,
```

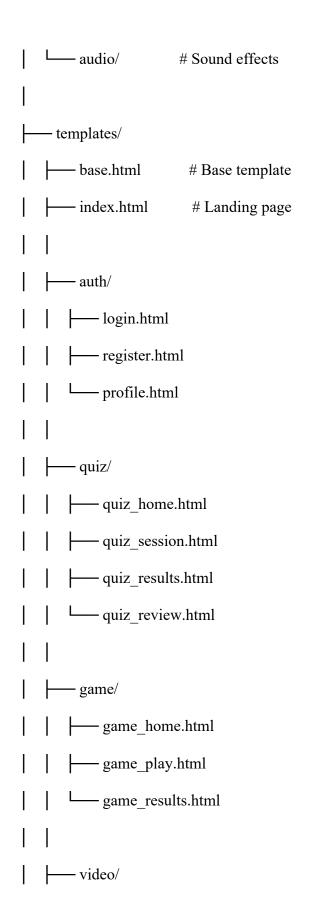
```
FOREIGN KEY (user_id) REFERENCES users(id)
```

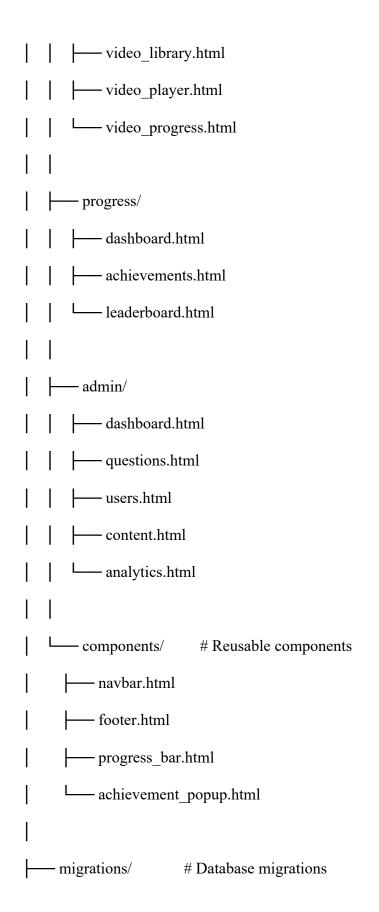
); driving-theory-app/ app.py # Main application file - config.py # Configuration settings - requirements.txt # Python dependencies # Application runner - run.py # Environment variables - .env - .gitignore - README.md - app/ - __init__.py # Initialize Flask app - models.py # SQLAlchemy models – forms.py # WTForms for validation — auth/ # Authentication blueprint — __init__.py - routes.py — forms.py — utils.py

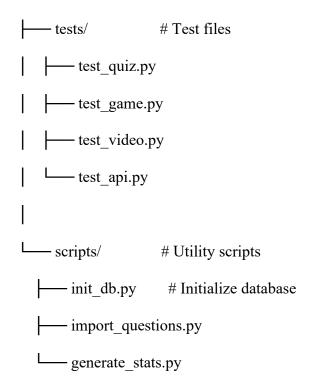
```
- main/ # Main pages blueprint
  init_.py
  routes.py
  └── utils.py
— quiz/ # Quiz functionality
 _____init___.py
  routes.py
 — utils.py
  api.py # API endpoints for AJAX
— game/ # Game functionality
 init_.py
  routes.py
  game_logic.py
  L—api.py
 — video/ # Video learning
  init_.py
  routes.py
  — utils.py
  L—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
L—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 scripts
   – game/
   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
```







3. Key Features to Implement

User Experience Features:

Gamification System

Points and XP system

Achievement badges

Daily streaks

Leaderboards

Technical Features:

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

Progressive Web App (PWA)

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