

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.

Database

-- 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