

# Flashcard Learning API

---

This is the backend API for a flashcard learning web application. It provides endpoints for user authentication, flashcard set management, topic handling, and AI-assisted flashcard generation.

## Features

- User authentication (register, login, logout)
- Flashcard set CRUD operations
- Topic management
- AI-assisted flashcard generation
- Protected and public routes

## Technologies Used

- Node.js
- Express.js
- Passport.js for authentication
- PostgreSQL (assumed based on typical usage with Sequelize)
- Sequelize ORM (assumed based on typical Express.js stack)

### ► API Routes

#### Authentication Routes

- `POST /api/auth/register`: Register a new user
- `POST /api/auth/login`: User login
- `POST /api/auth/logout`: User logout
- `GET /api/auth/protected`: Test protected route
- `GET /api/auth/get-users`: Get all users (likely admin only)

#### Flashcard Set Routes

- `GET /api/ai/flashcard-sets/:id`: Get a specific flashcard set
- `GET /api/ai/dashboard/flashcard-sets`: Get user's flashcard sets
- `POST /api/ai/flashcard-sets`: Create a new flashcard set with flashcards
- `PUT /api/ai/flashcard-sets/:id/update`: Update a flashcard set
- `DELETE /api/ai/flashcard-sets/:id`: Delete a flashcard set

#### Topic Routes

- `POST /api/ai/topics`: Create a new topic
- `GET /api/ai/topics/:id`: Get a specific topic
- `GET /api/ai/topics`: Get all topics
- `GET /api/ai/topics-dashboard/:id`: Get topic dashboard
- `DELETE /api/ai/topics/:id`: Delete a topic

#### Flashcard Routes

- `POST /api/ai/flashcards`: Generate new flashcards using AI
- `GET /api/ai/flashcards/:id`: Get a specific flashcard

## Setup and Installation

1. Clone the repository
2. Install dependencies: `npm install`
3. Set up environment variables in a `.env` file
4. Initialize the database
5. Start the server: `npm start`

## Environment Variables

- `NODE_ENV`: Set to 'production' for production environment
- `SERVER_PORT`: Port number for the server
- `CLIENT_URL`: URL of the frontend client for CORS
- Database connection details (specifics depend on your setup)

## Authentication

The API uses JSON Web Tokens (JWT) for authentication. Protected routes require a valid JWT to be included in the Authorization header of the request.

## Error Handling

The API includes centralized error handling middleware to process and return appropriate error responses.

## AI Integration

The API includes routes for AI-assisted flashcard generation. The specific AI model and integration details should be implemented in the `aiController`.

## Contributing

[Include guidelines for contributing to the project]

## License

[Specify the license under which this project is released]