EduStreak Project Overview

Project Name

EduStreak

Description

EduStreak is a modern Learning Management System (LMS) that promotes daily engagement through a streak-based reward mechanism. The platform enables users to register, log in, track their learning streaks, and maintain consistency in their learning journey.

Key Features

- User Authentication (Register/Login)
- Streak Tracker (Increases if user is active daily)
- Protected Routes using JWT
- RESTful APIs with Express
- MongoDB for Persistent Storage
- Frontend built with React
- Responsive UI with Tailwind CSS
- Streak Update Logic on Login

Tech Stack

Frontend: React, Tailwind CSS

Backend: Node.js, Express

Database: MongoDB

Other Libraries

- JWT: Used to create and verify JSON Web Tokens for secure user authentication and authorization.
- Bcrypt: For hashing passwords before storing them in the database to ensure user security.
- Axios: Handles HTTP requests from the React frontend to the backend APIs.
- Concurrently: Runs both the frontend and backend servers together during development for efficiency.

EduStreak Project Overview

Approach to Development

1. Planning:

- Defined key features like streak logic, authentication, dashboard UI.

- Created basic wireframes and data flow.

2. Backend Development:

- Created user model with fields: name, email, password, streak, lastActiveDate.

- Built user registration and login endpoints with JWT token generation.

- Implemented streak update logic based on user login date.

3. Frontend Development:

- Set up routes for Register, Login, and Dashboard.

- Designed StreakTracker component to display user streak visually.

- Used Axios for API requests and protected dashboard using cookies.

4. Testing & Debugging:

- Tested with different user scenarios to verify streak logic.

- Handled edge cases like repeated logins on the same day.

5. Deployment (optional):

- Ready to deploy on platforms like Vercel (frontend) and Render (backend).

Functionality Breakdown

Register:

- User inputs name, email, password.

- Backend creates user and returns a JWT token.

Login:

- On success, streak is checked:

  - If user was active yesterday, increment streak.

- If not active yesterday, reset streak to 1.
- If already active today, keep current streak.

Streak Tracker:

- Pulls current user streak and renders it visually.

Interviewer May Ask (with Answers)

- How does the streak logic work?

The backend compares the current date with the user's last active date. If it's the same day, the streak remains. If it was yesterday, the streak increases. If it's older, the streak resets to 1.

- How do you ensure security during login?

Passwords are securely hashed using Bcrypt. JWTs are stored in HttpOnly cookies to prevent XSS attacks. Protected routes validate tokens.

- What would happen if a user manipulates the system clock?

The server handles date calculations using its own clock, not the client?s. So even if users change their device's time, it won?t impact streak logic.

- How would you scale EduStreak for 100K users?

Use MongoDB indexing for fast access, deploy load balancers, implement Redis for caching, and use containerized services (like Docker) with Kubernetes for orchestration.

- How do you manage token expiration?

Tokens are issued with expiry time. Backend validates on every protected route. Refresh tokens and re-authentication can be added for long sessions.

- What improvements would you make next?
  - Social login via Google
  - Admin dashboard for managing users/content

EduStreak Project Overview

- Email reminders for streaks
- Gamification with badges/leaderboards

- How did you manage API errors on frontend?

Errors are caught using try-catch blocks in Axios requests. Messages are shown via alerts or toast notifications.

- What are the advantages of using Tailwind CSS?

Tailwind?s utility-first classes simplify styling without context switching. It provides responsive design tools and keeps code clean and scalable.

- What is the role of useEffect/useState in your app?

useState stores component states like form inputs. useEffect handles side-effects such as fetching streak data on component load or login.

- Can you walk through your folder structure?

client/: React app
- components/: UI components (StreakTracker, Navbar)
- pages/: Register, Login, Dashboard
- utils/: Axios setup for API calls

server/: Express backend
- models/: User schema
- controllers/: Business logic
- routes/: API endpoints
- middlewares/: Auth & error handlers
- server.js: App entry point

GitHub Repository

https://github.com/dharmendra4522/EduStreak

EduStreak Project Overview

Video Demo

https://drive.google.com/file/d/1bZ...DemoLinkHere

Future Enhancements

- Add Learning Modules & Progress Bars
- Daily Email Reminders for Streak Maintenance
- Leaderboard & Gamification
- Social Login (Google Auth)
- Admin Panel for Content Management

Created by

Dharmendra Vishvkarma - April 2025