Project Blueprint: PathPilot — Personalized Learning Path Platform

---

🔎Overview

PathPilot is a full-stack, AI-enhanced web platform that provides personalized learning roadmaps based on user skill levels. It aims to address the inefficiencies in one-size-fits-all online learning by creating adaptive paths and tracking progress with gamification, powered by the MERN stack and AI models.

🔧Core Technologies

- **Frontend:** React.js, Tailwind CSS, Redux Toolkit
- **Backend:** Node.js, Express.js
- **Database:** MongoDB Atlas
- **Authentication:** JWT + Google OAuth
- **AI Services:** OpenAI API / Custom ML model (Python Flask microservice)
- **APIs:** YouTube Data API, Medium RSS API
- **Others:** GraphQL, Redis (for caching leaderboards), Docker, Swagger (API docs)

🩲Features

1. User Authentication

- Secure signup/login (email + password)
- Google OAuth
- Role-based access (Learner, Admin, Mentor)

2. Skill Assessment Engine

- Initial quiz per topic (JS, React, etc.)
- Score-based skill classification (Beginner, Intermediate, Advanced)
- AI feedback on weak areas (using OpenAI API)

3. Personalized Learning Path Generator

- Based on quiz results, generate a roadmap
- Path includes modules with pre-requisites and difficulty level
- Learner can reorder or skip modules

4. Module Content

- External resources (YouTube videos, Medium articles)
- Admin-curated and auto-fetched via APIs
- Quiz at the end of each module

5. Gamification System

- XP system for completing modules, quizzes, and daily logins
- Levels and badges
- Leaderboard with caching (Redis)

6. Progress Dashboard

- Interactive timeline view of learning path
- Visual progress bars and streak counters
- Resume from last completed module

7. Admin Panel

- Manage users, topics, modules, and quizzes
- Moderate content quality
- Analytics: user engagement, quiz success rate

8. API Layer

- REST API for CRUD operations
- GraphQL API for flexible frontend queries
- Swagger docs for API endpoints

📁Project Structure (Monorepo)

```
pathpilot/
├── client/              # React frontend
│   ├── src/
│   │   ├── components/
│   │   ├── pages/
│   │   ├── services/
│   │   └── App.jsx
├── server/              # Express backend
│   ├── routes/
│   ├── controllers/
│   ├── models/
│   ├── graphql/
│   └── server.js
├── ml-service/          # (Optional) Python AI microservice
├── .env
├── Dockerfile
└── README.md
```

🔔Deployment Plan

- **Frontend:** Vercel / Netlify
- **Backend:** Render / Railway / AWS EC2
- **Database:** MongoDB Atlas
- **CI/CD:** GitHub Actions (auto-deploy on push)

📊Sample MongoDB Collections

users

```
{
  "_id": "abc123",
  "email": "john@example.com",
  "name": "John Doe",
  "role": "learner",
  "xp": 350,
  "skills": {
    "JavaScript": "intermediate",
    "React": "beginner"
  }
}
```

learning_paths

```
{
  "userId": "abc123",
  "topics": [
    { "name": "JavaScript Basics", "status": "completed" },
    { "name": "React Components", "status": "in-progress" }
  ]
}
```

resources

```
{
  "topic": "React",
  "videos": [
    { "title": "React Crash Course", "url": "https://youtube.com/..." }
  ],
  "articles": [
    { "title": "Understanding Components", "url": "https://medium.com/..." }
  ]
}
```

💡FAANG-Level Add-ons (Optional but Impressive)

- Microservice for AI quiz feedback (Python Flask)
- PDF certificate generation (PDFKit)
- Peer learning forum (simple comment threads per module)
- Offline support via PWA

📆Development Timeline (6 Weeks)

Week	Tasks
1	Setup project, GitHub repo, auth (JWT + Google)
2	Design DB schema, build skill assessment & quiz system
3	Implement dynamic learning path logic
4	Connect YouTube/Medium APIs, build module pages
5	Gamification + progress tracking UI
6	Final testing, deploy, polish UI, write README

📔Final Deliverables for Resume / GitHub

- Live Demo URL
- 2-min walkthrough video
- Clean, illustrated README
- GitHub project with commit history
- Swagger API documentation
- Architecture diagram

Outcome: A highly impressive full-stack AI-powered learning platform that demonstrates your frontend, backend, API design, authentication, database modeling, and AI integration skills — perfect for standing out in FAANG internship interviews.