Project Proposal: PathFinder AI

An AI-Powered Visual Learning Path Generator

Proponent: Khadija Ismail Goni
Course: Full Stack Development Final Project

1. Abstract

PathFinder AI is a full-stack MERN (MongoDB, Express.js, React, Node.js) application designed to address **Sustainable Development Goal 4 (SDG 4): Quality Education**.

The project begins with a **public-facing landing page** that clearly articulates its value, inviting users to sign up. The core application transforms complex educational topics into simple, interactive, and visual mind maps.

By leveraging the **Google Gemini API** to generate structured content and the **React Flow** library to render dynamic graphs, PathFinder AI provides an inclusive learning tool.

The application operates on a **freemium model**, with subscription payments managed by **Paystack** to ensure a seamless and regionally-relevant monetization structure.

2. Problem Statement

A primary barrier to equitable education is not just access to information, but its *digestibility*. Many students, especially in self-directed learning, face an “information wall” when tackling complex subjects.

This is ineffective for visual learners, students with learning disabilities, or non-native speakers. This "learning gap" undermines inclusive education.

Furthermore, many students hesitate to sign up for a tool they don’t yet understand, and high-quality educational platforms often lack a sustainable business model, limiting their long-term impact.

3. Proposed Solution

PathFinder AI is a web-based tool that functions as an “AI concept mapper” with a clear user flow and a two-tier structure:

1. **Discovery (Landing Page):** A public, non-authenticated page that explains *what* the app does, *how* it works, and *why* it’s valuable. It will feature pricing tiers and clear calls-to-action (CTAs) to register or log in.

2. **Registration/Login:** The user signs up for the default **Free Plan**.

3. **Generation:** Once in the app, the user provides a complex topic (e.g., *“The French Revolution”*). The Node.js backend sends this topic to the **Google Gemini API**, which returns a structured **JSON object** representing a learning path.

4. **Visualization:** The React frontend uses **React Flow** to render an interactive mind map.

5. **Interaction:** The user can click any node to access AI-powered learning tools (some premium).

6. **Monetization (Paystack):** Free users have usage limits but can upgrade via a **Paystack-powered** checkout.

---

4. SDG Alignment (SDG 4: Quality Education)

This project targets SDG 4 by promoting **inclusive and equitable quality education**:

- **Target 4.3 (Equitable Access):** The **Free Tier** and **public landing page** ensure accessibility for everyone, lowering the financial barrier.

- **Target 4.5 (Inclusivity):** The visual format and **AI “ELI5” feature** support diverse learning styles.

- **Target 4.4 (Skills for Employment):** Premium tools like “Translate” and “Explain Deeper” help learners understand complex topics across languages.

---

5. Key Features & Monetization Structure

Free Tier (The “Access” Plan)

- **AI Mind Map Generation:** Up to **5** maps/month.
- **Interactive Learning:** Access to **ELI5 (Explain Like I'm 5)**.
- **Full Visual Interface:** Full React Flow experience.

Premium Tier (The "Mastery" Plan)

- **Unlimited** AI Map Generations.
- **Full Learning Module:**
    - "ELI5 (Explain Like I'm 5)"
    - **"Explain Deeper"** advanced, academic explanations
    - **"Translate"** multilingual support
- **Personalized Dashboard:**
    - Save and manage unlimited maps
    - Delete saved maps
- **Payment Integration:** Secure Paystack checkout.

---

6. Technical Architecture

Component	Technology	Purpose
Frontend UI	React.js	Dynamic, responsive interface
Routing	React Router	Manage public and protected routes
Graph Visualization	React Flow	Render interactive maps
Backend API	Node.js + Express.js	REST API, authentication, payment proxy
Database	MongoDB	Store users, maps, and payment data
AI Engine	Google Gemini API	Generate structured mind maps and text

Authentication	JSON Web Token (JWT)	Secure access control
Styling	Tailwind CSS	Modern “Glassmorphism” UI
Payment	Paystack	Handle subscriptions and verify payments
Deployment	Render	Host both frontend and backend

---

7. UI/UX Design: Glassmorphism

The design will use **Glassmorphism** blurred, frosted-glass panels with soft shadows and transparency.

This style provides a sleek, modern look for the **landing page**, creating a strong first impression.

It’s simple to achieve with Tailwind CSS, allowing more time for functionality development. The UI will also visually differentiate between **free** and **premium** features with “Upgrade” prompts.

8. Database Schema (MongoDB)

User Schema

```
{
  username: String,
  email: { type: String, required: true, unique: true },
  password: { type: String, required: true },
  plan: { type: String, enum: ['free', 'premium'], default: 'free' },
  mapCount: { type: Number, default: 5 },
  paystackCustomerCode: { type: String, unique: true, sparse: true }
}
```

MindMap Schema

```
{
  author: { type: mongoose.Schema.Types.ObjectId, ref: 'User' },
  topic: { type: String, required: true },
  mapData: { type: Object, required: true },
  createdAt: { type: Date, default: Date.now }
}
```

---

9. API Routes

- `/api/auth/register`
- `/api/auth/login`
- `/api/maps/me`, `/api/maps/save`, `/api/maps/:id`
- `/api/ai/generate-map`
- `/api/ai/explain`, `/api/ai/translate`, `/api/ai/simplify`
- `/api/payments/initialize-transaction`
- `/api/payments/verify-transaction`

---

10. Project Roadmap (3 Weeks)

Week 1: Foundation & Public Pages

- Set up MERN stack, schemas, and JWT authentication
- Build **Landing Page**, **Login**, and **Register** pages
- Create `/api/ai/generate-map` and test with Postman
- ✅ Goal: User can register, log in, and access a basic dashboard

Week 2: Core App & Tier Logic

- Build main dashboard with React Flow
- Connect AI generation route to frontend
- Add usage limits and protect premium routes

- Lock premium UI features with upgrade prompts

Week 3: Paystack Integration & Deployment

- Set up Paystack API keys
- Build pricing section and “Upgrade” button
- Implement `/initialize-transaction` and `/verify-transaction` routes
- Add callback page for payment verification
- Update `User.plan` to “premium” after successful payment
- Deploy to **Render** and test full flow

11. Deliverables

1. A **live, public Render deployment** of PathFinder AI
2. A **GitHub repository** with well-documented code
3. A **demo video** showing:
    - Landing → Sign Up → Generate → Upgrade → Premium Access