FYP Execution Plan (SDLC-Based)

> **Goal:** Build a supervisor-approved, industry-relevant FYP using AI integration while matching an absolute-beginner-to-growing developer skill level.

Phase 0: Mindset & Constraints (Very Important)

- This project is **not** about building everything perfectly.
- This project **is** about showing:
  - clear thinking
  - correct system design
  - controlled AI usage
  - working demo

Rule: If something feels too complex we simplify, not quit.

Phase 1: Requirements Analysis (Week 1)

Objective

Understand **what to build** and **what NOT to build**.

Functional Requirements

The system must:

- Allow user signup/login
- Allow skill input (guided, structured)
- Generate a learning roadmap
- Show tasks step-by-step
- Track progress

- Store user-specific data
- Use AI only for explanations & feedback

The system will NOT:

- Allow free AI chat
- Support unlimited skills in demo
- Auto-skip learning steps

Non-Functional Requirements

- Simple UI
- Beginner-friendly logic
- API cost controlled
- Stable demo over feature-rich demo

Deliverables

- Final requirements list (locked)
- Approved feature scope

Phase 2: System Design (Week 1–2)

2.1 High-Level Architecture

- Frontend: React
- Backend: Node.js + Express
- Database: MongoDB
- AI: LLM API (as a service)

Design principle:

> System is boss, AI is helper.

2.2 Data Design (Simple)

Core entities:

- User
- Skill
- Roadmap
- Module
- Task
- Progress

No over-normalization. Keep it readable.

2.3 Roadmap Skeleton Design

Roadmap stored as JSON:

- Levels
- Modules
- Tasks
- Rules (pass/fail)

AI fills content **inside** these slots.

Phase 3: UX & UI Design (Week 2)

Objective

Design before coding.

Screens to Design:

- Login / Signup
- Skill Input Form
- Roadmap View
- Task View
- Progress Dashboard

Tools

- Pen & paper OR Figma
- No animations needed

Deliverables

- Basic UI wireframes

Phase 4: Development (Week 3–5)

> Development starts ONLY after design clarity.

4.1 Backend Development

- Auth APIs
- Skill submission API
- Roadmap generation API
- Progress save API

Start with dummy data, then add AI.

4.2 AI Integration (Controlled)

AI used only when:

- roadmap content needed
- user feedback needed

Prompt structure fixed.

AI output saved in DB.

4.3 Frontend Development

- Forms
- Conditional rendering

- API integration
- Progress states

No complex state management required.

Phase 5: Testing (Week 6)

Testing Types

- Manual testing
- User flow testing
- AI response sanity check

Focus: demo stability.

Phase 6: Deployment (Week 6)

- Backend: Render / Railway
- Frontend: Netlify / Vercel
- Environment variables secured

Phase 7: Documentation & Viva Prep (Final Week)

Documentation

- System overview
- AI role explanation
- Screenshots
- Limitations

Viva Focus

- Why AI is controlled
- What problem is solved

- What was intentionally excluded

Skill-Level Alignment (Reality Check)

This roadmap assumes:

- JS basics improving
- MERN learning ongoing
- Feature scope limited

Result: A complete, defendable FYP — not burnout.

Final Note

This execution plan is designed to:

- protect you from overengineering
- match learning speed
- impress supervisors
- build real confidence

You are not late. You are aligned.