StudyPilot PRODUCT README1

Product Requirements

Title:

StudyPilot – A Smart Syllabus Organizer and Study Planner

🎯 Goal:

To create a web-based academic productivity tool that enables students to easily upload, organize, and access departmental syllabi by semester, while dynamically generating personalized study schedules, reminders, and tracking academic progress.

🧰 Core Technologies:

Component	Tech Stack
Frontend	Next.js, TypeScript, Tailwind CSS, Axios
Backend	Java Spring Boot (Maven), Apache Commons FileUpload, Apache PDFBox
Deployment	Render (for Java backend), Vercel (for frontend)
Database	PostgreSQL (via Supabase or hosted manually)
Storage	Supabase (for PDFs and related metadata)
API Communication	RESTful (JSON-based) via Axios

📦 Core Features:

1. **Syllabus Upload & Management**
    - Upload PDF syllabi with metadata (course title, department, semester).
    - Parse syllabus using Apache PDFBox.
    - Categorize and store in Supabase/Postgres with search functionality.

2. **Dynamic Study Plan Generation**
    - Analyze syllabus structure (number of topics, pages, or chapters).
    - Use algorithms to auto-distribute content over available weeks.
    - Allow manual adjustments based on user availability or pace.
3. **Scheduling & Reminders**
    - Weekly or daily schedule generation.
    - Notifications/reminders via email or local browser alerts.
    - Optional calendar sync (future).
4. **User Dashboard**
    - Visual representation of progress (e.g., topics completed).
    - Upcoming sessions/reminders.
    - Statistics (daily, weekly, course-wise).
5. **Landing Page**
    - Sleek, informative homepage for new users.
    - Login/register, feature highlights, quick start.

🔁 App Flow

```
[Landing Page]
  ↓
[User Auth/Login/Register]
  ↓
[Dashboard]
  ↳ Upload Syllabus (PDF)
    ↓
  ↳ PDFBox parses metadata (title, semester, course content)
    ↓
  ↳ Store PDF + Metadata in Supabase
    ↓
```

```
    ↳ Generate Study Plan
        ↳ User inputs: exam date, study hours/day
        ↳ Algorithm distributes content → Calendar
        ↳ Reminders & sessions generated
    ↳ Dashboard tracks completion + upcoming tasks
```

🧠 Study Plan Algorithm – Logic Design

Inputs:

- Syllabus topics (parsed or manually entered)
- Total available weeks or end date (exam date)
- Daily/weekly available study time

Algorithm:

```
totalTopics = parsedTopics.size();
daysAvailable = ChronoUnit.DAYS.between(LocalDate.now(), examDate);
topicsPerDay = Math.ceil((double) totalTopics / daysAvailable);

// For each day, assign topicsPerDay topics
List<StudyDay> plan = new ArrayList<>();
for (int i = 0; i < daysAvailable; i++) {
    LocalDate date = today.plusDays(i);
    List<Topic> assigned = getNextTopics(topicsPerDay);
    plan.add(new StudyDay(date, assigned));
}
```

Optional Enhancements:

- Weighted topic distribution (e.g., based on difficulty/duration).
- Skipping weekends or unavailable days.
- Buffer days for revision.

🔐 Backend API Design (Spring Boot)

Method	Route	Description
`POST`	`/api/syllabus/upload`	Upload PDF and metadata
`GET`	`/api/syllabus/{id}`	Fetch syllabus data
`POST`	`/api/study-plan/generate`	Generate study plan
`GET`	`/api/study-plan/{userId}`	Get current user plan
`PUT`	`/api/progress/{dayId}`	Mark a day/lesson as complete

🖥️ Frontend Structure (Next.js)

- `/`: Landing Page
- `/dashboard`: Main user interface
- `/upload`: Upload syllabus
- `/planner`: View & edit study plan
- `/progress`: Progress tracking view
- `/api/*`: Axios calls to Spring Boot

📐 UI Design Principles

- **Landing Page**: Hero section, CTA, features grid
- **Dashboard**: Modular cards (progress, reminders, next session)
- **Study Plan**: Calendar (FullCalendar or custom), topic timeline
- **Visuals**: Clean, official UI (white + blue scheme), consistent spacing, animations with Framer Motion

---

🔧 Recommended Libraries & Tools

- **Frontend**:
  - `FullCalendar`, `Chart.js`, `TailwindCSS`, `Framer Motion`

- **Backend**:
  - `Apache PDFBox`, `Commons FileUpload`, `Spring Scheduler`
- **Database**:
  - `JPA/Hibernate`, `Supabase SDK`
- **Other**:
  - `Quartz` (optional for reminder scheduling)
