

CAMPUS SYSTEM — SIMPLE FULL DOCUMENT

(Python + React)

This document is the **complete, simple blueprint** for building the CAMPUS system using: - **Python (FastAPI)** → Backend API - **React** → Frontend - **PostgreSQL** → Database

No code included — only structure so you can build everything.

1. SYSTEM MODULES (English Names)

Core Student System

- Student Accounts
- Authentication & Roles
- Student Profiles
- Academic Calendar
- Programs / Majors / Semesters
- Module & Course Management
- Groups & Sections

Learning System (LMS)

- Courses
- Materials (PDFs, slides...)
- Assignments
- Quizzes
- Attendance

Administrative Services

- Certificates Requests
- Administrative Requests
- Document Downloads
- Appointment Booking

Exams System

- Exam Schedule
- Exam Rooms & Seating
- Convocations
- Grades
- Module Validation
- Rattrapage

Communication

- Announcements
- Notifications
- Messaging (optional)

Admin Panel

- Student Management
 - Teacher Management
 - Program Structure
 - Rooms Management
 - Exam Planning
 - Dashboard & Statistics
-

2. SYSTEM ARCHITECTURE (Python + React)

Frontend: React

- React + Vite
- React Router
- Axios for API
- State: Zustand or Redux (optional)

Backend: FastAPI

- REST API
- Authentication: JWT
- ORM: SQLAlchemy
- File uploads: FastAPI built-in
- Background tasks (emails, PDF)

Database: PostgreSQL

- Students
- Courses
- Teachers
- Grades
- Requests
- Exams

Storage:

- Files stored in: Local folder or S3

Optional Services:

- Python micro-service for PDF
 - Python micro-service for AI assistant
-

3. DATABASE DESIGN (TABLES + RELATIONS)

Users Table

- id
- email
- password
- role (student / teacher / admin)

Students Table

- id
- user_id (FK → users)
- CNE
- CIN
- birthdate
- major_id
- semester

Teachers Table

- id
- user_id
- department

Majors (Filières)

- id
- name

Semesters

- id
- name
- major_id

Modules (Subjects)

- id
- name
- semester_id
- professor_id

Courses Materials

- id
- module_id
- title
- file_url

Announcements

- id
- title
- content
- target (all / semester / major)

Requests (Administrative)

- id
- student_id
- type
- status
- generated_pdf_url

Exams

- id
- module_id
- date
- room

Grades

- id
- student_id
- module_id
- grade

4. UI / UX STRUCTURE (WIREFRAMES SIMPLIFIED)

Student Dashboard

- Profile summary
- Latest announcements
- Today schedule
- Quick actions: certificates, materials, grades

Left Menu

- Dashboard
- Courses
- Materials
- Exams
- Grades
- Requests
- Settings

Admin Dashboard

- Students list
 - Teachers list
 - Modules
 - Exams planning
 - Requests panel
 - Statistics
-

5. ROADMAP — STEP BY STEP

Phase 1 — Setup (1-2 days)

1. Create FastAPI project
2. Connect PostgreSQL
3. Create tables using SQLAlchemy
4. Create basic CRUD for:
5. Users
6. Students
7. Majors
8. Modules

Phase 2 — Authentication (1 day)

1. JWT login
2. Roles: student / teacher / admin
3. Protect routes

Phase 3 — React Frontend (2-5 days)

1. React setup
2. Login page
3. Student dashboard
4. Admin dashboard

5. API integration

Phase 4 — Student Services (5–10 days)

1. Courses & materials
2. Announcements
3. Requests system
4. Certificates generation

Phase 5 — Exams Module (5–7 days)

1. Exam schedule
2. Rooms & convocation generation
3. Grades input
4. Grades display

Phase 6 — Polishing (3–5 days)

- Responsive design
- Better UI
- Notifications
- Performance

6. FINAL RESULT

A complete **Campus System** using: - **Python (FastAPI)** backend - **React** frontend - **PostgreSQL** database

Simple, clean, scalable.

Ask me if you want: - PDF version - More tables - Diagrams - Next.js alternative