

Campus Navigation Navigator - Complete Project Prompt

Project Overview

Build a comprehensive **Campus Information & Navigation System** called "Campus Connect" for Yenepoya Institute of Technology. This is a full-stack web application that integrates AI-powered assistance, academic resource management, student performance tracking, career development tools, and real-time campus information to enhance the overall campus experience for students, faculty, and administrators.

Core Technology Stack

Frontend

- **Framework:** React 18.3+ with TypeScript
- **Build Tool:** Vite 5.4+
- **UI Library:** shadcn-ui (Radix UI components)
- **Styling:** Tailwind CSS with custom campus theme
- **State Management:** React Query (@tanstack/react-query)
- **Routing:** React Router DOM v6
- **Forms:** React Hook Form with Zod validation
- **Charts:** Recharts for data visualization
- **Icons:** Lucide React
- **PDF Generation:** jsPDF, html2canvas, react-to-print
- **AI Integration:** @google/generative-ai (Gemini API)

Backend Services (Python Flask)

- **Framework:** Flask with CORS support
- **AI/ML:** Google Generative AI (Gemini), LangChain
- **Database:** Firebase (Firestore, Authentication, Storage)
- **File Processing:** pandas, openpyxl for CSV/Excel handling
- **Vector Database:** FAISS for RAG (Retrieval Augmented Generation)
- **API Framework:** FastAPI (for csv_rag_backend)

Development Ports

- Frontend (Vite): **5173**
- API Backend: **5000**

- Auth Backend: **5001**
- CSV RAG Backend: **5002**
- Course Backend: **5003**
- Live Events Backend: **5004**
- Study Materials Backend: **5005**
- Exam Hall Backend: **5006**
- Geolocation Backend: **5007**
- AI YouTube Search Backend: **5008**
- Student Performance Backend: **5009**
- Notifications Backend: **5010**
- AI Study Buddy Backend: **5011**
- Mock Interview Backend: **5012**
- Semester Results Backend: **5013**
- Digital Notice Board Backend: **5014**
- Resume Backend: **3001**
- Job Search Backend: **3002**

Complete Feature List

1. ****AI Campus Chatbot**** ■

Route: /chatbot

Backend: csv_rag_backend (Port 5002)

Features:

- Natural language query processing using Gemini AI
- RAG-based document retrieval from campus CSV data
- Student and faculty information search
- Campus facilities and services information
- Question paper and academic resource queries
- Contextual conversation history
- Real-time responses with streaming support

Implementation:

- Uses FAISS vector database for semantic search
- LangChain for document processing and retrieval
- CSV embeddings for student/faculty data
- Integration with campus knowledge base

2. ****Study Materials Management**** ■

Route: /study-materials

Admin Route: /admin/study-materials

Backend: study_materials_backend (Port 5005)

Features:

- Upload and download study materials (PDFs, documents, images)
- Category-based organization (by subject, semester, course)
- Search functionality for materials
- File preview capabilities
- Access control and permissions
- Version tracking
- Admin panel for content moderation

Implementation:

- Firebase Storage for file hosting
- Firestore for metadata storage
- File type validation and size limits
- Organized folder structure by department/course

3. ****Course Information System**** ■

Route: /courses

Admin Route: /admin/courses

Backend: course_backend (Port 5003)

Features:

- Comprehensive course catalog
- Course details (syllabus, credits, prerequisites)
- Faculty assignment to courses
- Course schedule and timings
- Material distribution per course
- YouTube tutorial recommendations
- Course enrollment tracking

Implementation:

- Firestore database for course data
- CRUD operations for admin
- Student view with filtering and search

- Integration with study materials

4. ****Live Events & Announcements**** ■

Route: /events

Admin Route: /admin/events

Backend: live_events_backend (Port 5004)

Features:

- Real-time event calendar
- Event creation and management
- Event registration and RSVP
- Location and timing details
- Event notifications
- Past events archive
- Event categories (academic, cultural, sports, etc.)

Implementation:

- Firestore for event storage
- Real-time updates using Firebase listeners
- Date-based filtering and sorting
- Admin dashboard for event management

5. ****Exam Hall Finder**** ■■

Route: /exam-hall-finder

Admin Route: /admin/exam-hall

Backend: exam_hall_backend (Port 5006)

Features:

- Search exam hall by student USN (University Seat Number)
- Automated seating allocation
- CSV upload for bulk student data
- Block, room, and seat number assignment
- Downloadable seating arrangements
- Template download for data upload
- Data validation and error handling

Implementation:

- CSV parsing with pandas

- Firestore for seating data storage
- Search by USN with instant results
- Admin bulk upload interface
- Export functionality for printing

6. ****Resume Builder**** ■

Route: /resume-builder

Backend: resume_backend (Port 3001)

Features:

- Interactive resume creation wizard
- Multiple professional templates
- Sections: Personal info, education, experience, skills, projects, certifications
- Real-time preview
- PDF download functionality
- Save and edit drafts
- Print-ready formatting

Implementation:

- React Hook Form for data collection
- jsPDF for PDF generation
- html2canvas for template rendering
- Local storage for draft saving
- Responsive template designs

7. ****Job Search Portal**** ■

Route: /job-search

Backend: job_search_backend (Port 3002)

Features:

- Job listings aggregation
- Filter by location, company, role, experience
- Job details and application links
- Bookmark favorite jobs
- Application tracking
- Career resources and tips
- Company information

Implementation:

- External job API integration
- Firestore for bookmarks and user preferences
- Search and filter functionality
- Responsive job cards with details

8. ****AI-Powered YouTube Search**** ■

Route: /youtube-search

Backend: ai_youtube_search_backend (Port 5008)

Features:

- AI-enhanced search for educational videos
- Topic-based recommendations
- Subject-specific tutorial search
- Video metadata display (title, channel, duration)
- Direct YouTube links
- Curated playlists for courses

Implementation:

- YouTube Data API integration
- Gemini AI for query enhancement
- Relevance-based sorting
- Thumbnail and preview display

9. ****AI Study Buddy**** ■

Route: /study-buddy

Backend: ai_study_buddy_backend (Port 5011)

Features:

- Personalized study plan generation
- Practice questions and quizzes
- Concept explanations and summaries
- Study schedule recommendations
- Topic-wise learning paths
- Progress tracking
- Adaptive learning suggestions

Implementation:

- Gemini AI for content generation
- User profile and learning history
- Interactive chat interface
- Subject and topic selection
- Markdown rendering for formatted responses

10. ****AI Mock Interviewer**** ■

Route: /mock-interview

Backend: mock_interview_backend (Port 5012)

Features:

- Voice-based mock interviews
- AI-generated interview questions
- Real-time speech recognition
- Answer evaluation and feedback
- Multiple interview types (technical, HR, behavioral)
- Performance scoring
- Improvement suggestions
- Interview history and analytics

Implementation:

- Web Speech API for voice recognition
- Gemini AI for question generation and evaluation
- Text-to-speech for AI interviewer
- Session recording and playback
- Detailed feedback reports

11. ****Student Performance Analyzer**** ■

Route: /performance-analyzer (Admin)

Routes: /student-upload, /select-semester, /select-internals

Backend: student_performance_backend (Port 5009)

Features:

- Internal marks entry and tracking
- Semester-wise performance analysis
- Subject-wise grade visualization
- Assignment marks management
- Performance trends and graphs

- Class average comparison
- Toppers list generation
- Export reports to PDF/Excel

Implementation:

- Firestore for marks storage
- Recharts for data visualization
- CSV upload for bulk entry
- Statistical analysis (mean, median, percentiles)
- Multi-semester comparison

12. ****Semester Results Management**** ■

Route: /upload-semester-result (Admin)

Backend: semester_results_backend (Port 5013)

Features:

- Semester result upload
- Student result viewing
- Subject-wise marks display
- SGPA/CGPA calculation
- Result analytics
- Pass/fail status tracking
- Downloadable mark sheets

Implementation:

- Excel/CSV parsing for bulk upload
- Firestore for result storage
- Automated GPA calculations
- Student-specific result retrieval
- Print-ready result format

13. ****Digital Notice Board**** ■

Route: /digital-notice-board

Admin Route: /admin/digital-notice-board

Backend: digital_notice_board_backend (Port 5014)

Features:

- Digital notice posting

- Category-based notices (academic, administrative, urgent)
- Notice expiry and archiving
- File attachments support
- Priority/urgent notice highlighting
- Search and filter notices
- Admin moderation

Implementation:

- Firestore for notice storage
- Firebase Storage for attachments
- Real-time notice updates
- Date-based sorting and filtering
- Rich text formatting support

14. ****Class Toppers Dashboard**** ■

Route: `/class-toppers` (Admin)

Features:

- Top performers by semester
- Subject-wise toppers
- Overall class ranking
- Performance metrics
- Visual leaderboards
- Filtering by department/batch

Implementation:

- Data aggregation from performance backend
- Sorting and ranking algorithms
- Visual charts and tables
- Export functionality

15. ****Admin Dashboard**** ■■■

Route: `/admin-dashboard`

Login Route: `/admin/login`

Features:

- Centralized admin control panel
- Quick access to all admin features

- System statistics and analytics
- User management
- Content moderation
- Notification settings
- Subject management
- Batch and department management

Sub-routes:

- /admin/chatbot-gp - Chatbot & Question Papers
- /admin/study-materials - Study Materials Management
- /admin/events - Events Management
- /admin/courses - Course Management
- /admin/subjects - Subject Management
- /admin/exam-hall - Exam Hall Management
- /admin/notifications - Notification Settings
- /admin/digital-notice-board - Notice Board Management

16. ****User Authentication & Profiles**** ■

Routes: /login, /signup, /forgot-password, /user-profile

Backend: auth (Port 5001)

Features:

- Student/faculty registration
- Secure login with Firebase Auth
- Password reset functionality
- Role-based access control (Student, Faculty, Admin)
- User profile management
- Profile picture upload
- Department and batch assignment
- Session management

Implementation:

- Firebase Authentication
- JWT token-based sessions
- Protected routes with AuthCheck
- Admin privilege verification
- Profile data in Firestore

17. ****Campus Navigation & Maps**** ■■

Route: /navigation

Backend: geolocation_backend (Port 5007)

Features:

- Interactive campus map
- Building and facility locations
- Turn-by-turn navigation
- Points of interest
- Google Maps integration
- Search for locations
- Directions and routes

Implementation:

- Google Maps API
- Custom campus markers
- Geolocation services
- Route calculation

18. ****Notification System**** ■

Backend: notifications_backend (Port 5010)

Admin Route: /admin/notifications

Features:

- Push notifications for events
- Exam reminders
- Result announcements
- Notice board alerts
- Customizable notification preferences
- Email notifications
- In-app notification center

Implementation:

- Firebase Cloud Messaging
- Email service integration
- User notification preferences
- Scheduled notifications
- Real-time alerts

19. ****Search Functionality**** ■

Route: /search-results

Features:

- Global search across all content
- Search study materials, courses, events
- Student/faculty search
- Filtered results by category
- Recent searches
- Search suggestions

Implementation:

- Firestore queries with indexing
- Full-text search capabilities
- Debounced search input
- Result ranking by relevance

System Architecture

Frontend Structure

```
src/
  ■■■ components/           # Reusable UI components
  ■   ■■■ NavBar.tsx       # Navigation bar
  ■   ■■■ AuthCheck.tsx    # Authentication wrapper
  ■   ■■■ InfoCard.tsx     # Feature cards
  ■   ■■■ ui/              # shadcn-ui components
  ■■■ pages/               # Page components (40+ pages)
  ■■■ hooks/               # Custom React hooks
  ■■■ lib/                 # Utility functions
  ■■■ styles/              # Global styles
```

Backend Structure

```
campus-navigation-navigator/
  ■■■ api/                 # General API (Port 5000)
  ■■■ auth/                # Authentication (Port 5001)
  ■■■ csv_rag_backend/     # AI Chatbot (Port 5002)
  ■■■ course_backend/      # Courses (Port 5003)
  ■■■ live_events_backend/ # Events (Port 5004)
  ■■■ study_materials_backend/ # Study Materials (Port 5005)
  ■■■ exam_hall_backend/   # Exam Hall (Port 5006)
  ■■■ geolocation_backend/ # Navigation (Port 5007)
  ■■■ ai_youtube_search_backend/ # YouTube Search (Port 5008)
  ■■■ student_performance_backend/ # Performance (Port 5009)
  ■■■ notifications_backend/ # Notifications (Port 5010)
  ■■■ ai_study_buddy_backend/ # Study Buddy (Port 5011)
  ■■■ mock_interview_backend/ # Mock Interview (Port 5012)
  ■■■ semester_results_backend/ # Semester Results (Port 5013)
  ■■■ digital_notice_board_backend/ # Notice Board (Port 5014)
  ■■■ resume_backend/      # Resume Builder (Port 3001)
```

Design Requirements

Visual Design

- **Color Scheme:** Custom campus theme with primary colors
- Primary: Campus blue/teal gradient
- Accent: Vibrant complementary colors
- Background: Clean white/gray-50
- **Typography:** Modern, readable fonts (system fonts + Google Fonts)
- **Layout:** Responsive grid layouts, mobile-first approach
- **Animations:** Smooth transitions, fade-in effects, hover states
- **Cards:** Glassmorphism effects, shadow elevations
- **Hero Sections:** Full-width video/image backgrounds with overlays

User Experience

- Intuitive navigation with persistent navbar
- Quick access cards on homepage
- Search functionality in navbar
- Breadcrumb navigation for deep pages
- Loading states and error handling
- Toast notifications for user actions
- Responsive design for mobile, tablet, desktop

Key Implementation Details

Authentication Flow

User registers/logs in via Firebase Auth

JWT token stored in session

Protected routes check authentication status

4. Admin routes verify admin privileges

5. User profile data synced with Firestore

Data Flow

Frontend makes API calls to respective backend services

Backend validates requests and authenticates users

Backend queries Firebase/external APIs

4. Data processed and returned to frontend

5. Frontend updates UI with React Query caching

File Upload Flow

User selects file in frontend

File validated (type, size)

Uploaded to Firebase Storage

4. Metadata saved to Firestore

5. Download URL returned to user

AI Integration

User input sent to backend

Backend calls Gemini API with context

AI generates response

4. Response formatted and returned

5. Frontend displays with markdown rendering

Setup and Deployment

Frontend Setup

```
npm install
npm run dev
```

Backend Setup

```
# Create virtual environment
python -m venv .venv

# Activate (Windows)
.venv\Scripts\activate

# Install dependencies
pip install -r requirements.txt

# Start all backends
python start_all_backends.py
```

Environment Variables

- Firebase configuration (API keys, project ID)
- Gemini API keys for AI features

- YouTube API key
- Job search API credentials
- CORS allowed origins

Security Features

- Firebase Authentication with secure tokens
- Role-based access control (RBAC)
- Input validation and sanitization
- CORS configuration for API security
- File upload validation
- SQL injection prevention
- XSS protection
- Secure password handling

Future Enhancements

Mobile application (React Native)

Real-time chat between students

Library seat booking system

4. Campus food ordering

5. Lost & found system

6. Ride-sharing/carpool

7. Study group finder

8. Campus feedback system

9. Emergency alert system

10. Attendance tracker

Testing Requirements

- Unit tests for backend APIs
- Integration tests for frontend components
- End-to-end testing for critical flows
- Performance testing for AI features
- Security testing for authentication
- Cross-browser compatibility testing
- Mobile responsiveness testing

Documentation

- API documentation for all endpoints
- User guides for students and faculty
- Admin manual for system management
- Developer documentation for maintenance
- Deployment guide
- Troubleshooting guide

Success Metrics

- User adoption rate (students, faculty)
- Feature usage analytics
- System uptime and reliability
- Response time for AI features
- User satisfaction scores
- Content engagement (materials, events)
- Search effectiveness
- Mobile vs desktop usage

This comprehensive system serves as a one-stop solution for all campus-related information and services, leveraging modern web technologies and AI to create an exceptional user experience for the entire campus community.