# Business Requirement Document (BRD)

## Project Title: Student Travel LLM App

## Objective:

To develop a Student Travel LLM (Large Language Model) App aimed at helping students planning their travel to India for exams. The app will assist students in finding affordable lodging and food while also streamlining essential travel, academic, and administrative support.

## 1. Business Requirements

### 1.1 Identify Users & Needs:

* **Primary Users**: Students traveling for exams.
* **Secondary Users**: Educational institutions, transport providers, healthcare services, accommodation providers.
* **Needs**:
  + Affordable lodging & meal options.
  + Help with travel planning and documentation.
  + Easy form filling and document upload.
  + Travel loans and insurance.
  + Emergency and healthcare information.
  + Real-time application tracking and course presence.

### 1.2 Problems to Solve:

* High travel and stay cost for students.
* Unorganized travel and documentation process.
* Lack of information about affordable options.
* Manual processing delays.

### 1.3 Key Features:

* **Form Filing & Document Uploading**
* **Travel Loan Applications**
* **Document Checklist Generator**
* **Travel Itinerary Planner**
* **Location & Map Integration**
* **Pickup & Transportation Info**
* **Accommodation Listings**
* **University Info Integration**
* **Meal Plan Options**
* **Student Health Insurance**
* **Emergency Contacts Access**
* **Application Status Tracker** (In Process / Approved / Rejected)
* **Student Location Tracking**
* **Student Presence Status in Courses**

### 1.4 Existing Solution Research:

* Swiggy, Zomato, MakeMyTrip, and others offer partial solutions but lack an education-travel integrated app.

## 2. Feasibility Study

### 2.1 Data Availability:

* Student travel data, lodging databases, transport APIs, health and university records.

### 2.2 Legal/Compliance Review:

* GDPR and Indian Data Protection Bill compliance.
* Consent for location tracking and document sharing.

### 2.3 Resource & Budget Assessment:

* Budget for development, LLM integration, cloud hosting, and licensing.
* HR: Developers, LLM specialists, UI/UX, compliance, support.

### 2.4 Technical Feasibility:

* Integration with third-party APIs (e.g., IRCTC, Google Maps, PayTM)
* Cloud-native microservices-based deployment

## 3. Technical Architecture

### 3.1 Frontend & Backend Design:

* **Frontend**: Mobile-first (Android/iOS) and Web Interface
* **Backend**: RESTful APIs with GraphQL Support

### 3.2 Database Plan:

* **Student Data**
* **Document Storage (Encrypted)**
* **Accommodation and Travel Listings**
* **University & Course Data**
* **Application Status Tracking**

### 3.3 LLM Integration:

* Prompt-engineered GPT-based system
* Supports Natural Language Form Filling, Recommendations, and Query Resolution

### 3.4 Admin Panel:

* Dashboard for application review, student tracking, analytics, and communication

## 4. Technology Stack

### 4.1 Frontend:

* Flutter / React Native / Angular

### 4.2 Backend:

* Node.js / Django / FastAPI

### 4.3 Database:

* PostgreSQL / MongoDB / Firebase

### 4.4 LLM Provider:

* OpenAI GPT-4 / Claude / Custom RAG LLM

### 4.5 Hosting & Deployment:

* AWS / Azure / GCP (Kubernetes, CI/CD via GitHub Actions or GitLab CI)

## 5. Delivery Plan

### 5.1 Development Methodology:

* Agile (2-week sprints)

### 5.2 Version Control:

* Git (GitHub/GitLab)

### 5.3 Testing:

* Unit Testing, UAT, Load Testing, Accessibility Testing

### 5.4 Deployment:

* CI/CD Pipelines, Canary Deployments

### 5.5 MVP & Iterations:

* **MVP**: Basic itinerary, lodging, form upload, and application tracker.
* **V1**: Add loans, meal options, and student presence tracker.

## 6. Support Plan

### 6.1 User Support:

* In-app help center, chat, and ticketing system

### 6.2 Monitoring Tools:

* Prometheus, Grafana, Sentry

### 6.3 Feedback Collection:

* In-app surveys, ratings, and issue reporting

### 6.4 Documentation:

* Internal (Dev & Admin) + User Manuals

### 6.5 Maintenance:

* Monthly maintenance windows, patching plan, and hotfix process

## 7. High-Level Architecture Principles

* **Segregation**: Logical and role-based separation of services
* **API-First Approach**: All functionality exposed through secure APIs
* **Prompt Engineering Layer**: Custom prompts based on student profile and use case
* **Context Management**: Session-based context retention for continuity
* **Observability & Monitoring**: End-to-end observability stack
* **Security**: OAuth2, SSL, RBAC, data encryption, compliance with regulations
* **Scalability**: Cloud-native, autoscaling enabled
* **Fallbacks & Human-in-the-Loop (RAG)**: Escalation to human agents with audit trail
* **Model Versioning and Deployment**: Canary deployments, shadow testing
* **Ethical AI**: Bias monitoring, transparent use, opt-out for sensitive features
* **MCP (Model Control Plane)**: Governance, monitoring, and performance tracking of LLM components