□ TransLingua – Project Planning Phase

1□ Introduction to Planning Phase

The Planning phase is the foundation of the TransLingua project. It defines the project objectives, scope, requirements, resources, and development strategy. Proper planning reduces risks, avoids scope creep, and ensures smooth execution.

In this phase, we clearly identify:

- What problem we are solving
- Who the target users are
- What features the system must include
- Which technologies will be used

□ 2□ Project Objective

The main objective of TransLingua is:

- To develop a secure multilingual translation web application
- To allow users to translate text between different languages
- To store translation history securely
- To implement role-based authentication
- To design a scalable and maintainable backend architecture

□ 3□ Problem Statement

Communication barriers exist due to language differences. Many users require a simple, secure, and efficient platform to translate text for:

- Education
- Travel
- Business
- International communication

Existing tools may lack:

- Secure login systems
- Translation history management
- Role-based access control
- Custom backend integration

TransLingua aims to solve these limitations.

□ 4□ Stakeholder & User Identification

Primary Users:

- Students
- Travelers
- Business professionals
- Content creators

Secondary Users:

- Admins (system monitoring & user management)

□ 5□ Scope of the Project

✔ In Scope:

- User Registration & Login
- JWT-based Authentication
- Role-based Authorization (Admin/User)
- Text Translation Feature
- Translation History Storage
- REST API Architecture
- Database Integration (MySQL)

□ Out of Scope:

- Real-time voice translation (future enhancement)
- AI model training
- Offline mobile application

□ 6□ Technology Stack Selection

Layer	Technology
Frontend	HTML, CSS, JavaScript
Backend	Spring Boot
Security	Spring Security + JWT
Database	MySQL
ORM	JPA / Hibernate
Testing	Postman

Version Control	Git & GitHub

Why Spring Boot?

- Rapid development
- REST API support
- Strong security integration
- Scalable architecture

7 Functional Requirements

- User must be able to register.
- User must be able to log in.
- System must generate JWT token after login.
- User must be able to translate text.
- System must store translation history.
- Admin must manage users.
- Role-based endpoint restriction must be implemented.

⚙ 8 Non-Functional Requirements

Performance

- Fast API response time
- Efficient database queries

Security

- Password encryption using BCrypt
- JWT expiration handling
- Role-based access

Scalability

- Stateless authentication
- Cloud-deployable architecture

Maintainability

- Layered architecture
- Clean code practices

9 Risk Analysis

Risk	Mitigation
Security vulnerabilities	Use Spring Security best practices
Database overload	Index optimization
Token misuse	Short JWT expiration
API misuse	Role-based access control

□ □ Project Timeline (Sample)

Phase	Duration
Planning	1 Week
Design	1 Week
Development	2–3 Weeks
Testing	1 Week
Deployment	1 Week