Software Requirements Specification (SRS)

Project: English to Hindi Language Translator WebApp

Model: Helsinki-NLP / opus-mt-en-hi

1. Introduction

Purpose: This document defines the functional and non-functional requirements for the English to Hindi Language Translator Web Application.

Scope: The application translates English text into Hindi using an NLP model (opus-mt-en-hi). Backend is built with Python Flask and deployed on Render.

2. Overall Description

Product Perspective: Web-based application with frontend (HTML, CSS, JS), backend (Flask), NLP model integration, and cloud deployment on Render.

Product Functions:
- Accept English text input
- Process translation using NLP model
- Display Hindi output
- Handle errors (empty input/server error)

3. Functional Requirements

FR1: System shall accept English text input.
FR2: System shall process input using opus-mt-en-hi NLP model.
FR3: System shall display Hindi translation output.
FR4: System shall handle invalid or empty input.
FR5: System shall be publicly accessible after deployment on Render.

4. Non-Functional Requirements

Performance: Response time should be less than 3 seconds.

Security: Input validation and secure API handling.
Usability: Simple, responsive, mobile-friendly UI.
Reliability: 95% uptime on cloud platform.
Maintainability: Clean and modular code structure.

5. System Architecture

Architecture: Client-Server Model.

Components:
- Frontend (HTML/CSS/JS)
- Backend (Flask routes)
- NLP Model (Helsinki-NLP opus-mt-en-hi)
- Deployment (Render Cloud)

6. External Interface Requirements

User Interface:
- Textarea for input
- Translate button
- Output display section

API Endpoint:
POST /translate
Input: { "text": "Hello" }
Output: { "translation": "■■■■■■" }

7. Future Enhancements

- Voice to Text
- Hindi Text to Speech
- Translation History Storage
- Multi-language Support
- Dark Mode

8. Deployment Plan

1. Create GitHub repository
2. Add requirements.txt
3. Push code
4. Connect repository to Render
5. Set start command: gunicorn app:app
6. Deploy and test application