Software Requirement Specification (SRS)

Project: English to Hindi Language Translator Web Application

1. Introduction

1.1 Purpose

The purpose of this project is to develop a web-based Language Translator that translates English text into Hindi using the NLP model Helsinki-NLP/opus-mt-en-hi.

The system is developed using Python Flask and deployed on Render cloud platform.

1.2 Project Scope

The application accepts English text input, processes it using a pretrained NLP model, and displays translated Hindi output instantly through a web browser.

2. Overall Description

2.1 Product Perspective

Backend: Python Flask

Frontend: HTML, CSS, JavaScript

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

Deployment Platform: Render

2.2 Product Functions

- Accept English input text

- Process text using NLP model

- Return translated Hindi output

- Display translation on web interface

2.3 User Classes

General users who need English to Hindi translation.

2.4 Operating Environment

- Windows, Linux, Mac

- Chrome, Edge, Firefox browsers

- Python 3.9+

3. System Features

3.1 Translation Feature

Input: English text

Processing: Tokenization and model inference

Output: Hindi translated text

3.2 Error Handling

- Empty input validation

- Server error handling

- Model failure handling

4. External Interface Requirements

4.1 User Interface

- Text area for input

- Translate button

- Output display section

4.2 Software Interface

- Flask

- Transformers

- Torch

- Gunicorn

5. Non-Functional Requirements

5.1 Performance

Response time less than 5 seconds.

5.2 Security

HTTPS enabled. Input validation implemented.

5.3 Reliability

High uptime based on cloud hosting.

6. Deployment Requirements

Deployed on Render using GitHub repository, requirements.txt, Procfile, and Flask application.

7. Future Enhancements

- Speech-to-text integration

- Multi-language support

- User authentication and history

Conclusion

The system provides a simple and efficient English to Hindi translation web application using NLP model with Flask and cloud deployment.