

# Hindi Speech Transcription and Task Management Application Project Report

Sanskar

June 9, 2025

## Abstract

The Hindi Speech Transcription and Task Management Application is a Flask-based web platform designed to facilitate speech-to-text transcription and task assignment for administrative officers in Hindi-speaking regions. Integrated with Supabase for authentication and data storage, the application supports admin and officer roles, enabling task creation, assignment, and updates with designations from `officers.json`. This report outlines the project's objectives, architecture, features, development process, challenges, resolutions, and current status as of June 9, 2025.

## 1 Introduction

The Hindi Speech Transcription and Task Management Application aims to streamline administrative workflows by providing a platform for transcribing Hindi speech and managing tasks assigned to officers based on their designations. Developed as part of an internship at the National Informatics Centre (NIC), the project addresses the need for efficient task delegation and tracking in government settings, with a focus on Hindi-language support.

## 2 Project Objectives

- Develop a web application for Hindi speech transcription and task management.
- Support role-based access for admins (task creation) and officers (task updates).
- Integrate Supabase for secure authentication and data persistence.
- Use designations from `officers.json` for task assignments.
- Ensure Hindi text compatibility using Noto Sans Devanagari fonts.
- Deploy the application on Render for public access.

### 3 System Architecture

The application is built using a client-server architecture with the following components:

- **Frontend:** Flask templates (`index.html`, `transcripts.html`, `my_tasks.html`, `login.html`, `signup.html`) styled with Tailwind CSS (CDN) and Noto Sans Devanagari for Hindi text rendering.
- **Backend:** Flask 2.3.3, handling routes for authentication (`/login`, `/signup`, `/logout`), task management (`/save_transcript`, `/get_transcripts`, `/get_my_task`, `/update_task`, `/update_transcript`), and debugging (`/debug_officers`).
- **Database:** Supabase, hosting tables `tasks`, `task_assignments`, `user_roles`, and `user_designations` for task and user data.
- **Authentication:** Supabase Auth for email-based sign-up and login, with email verification disabled for testing.
- **Data Source:** `officers.json`, containing Hindi officer designations (e.g., “`राजेश कुमार अधिकारी` `अधीनस्थ अधिकारी` `अधीनस्थ अधिकारी`”).
- **Deployment:** Render, with Gunicorn as the WSGI server.

### 4 Features

- **Authentication:**
  - User sign-up with role (admin/officer) and designation selection.
  - Auto-login post-sign-up after disabling email verification.
  - Secure session management with Flask’s `session`.
- **Task Management:**
  - Admins create tasks via `/save_transcript`, assigning to specific officers or all designations.
  - Officers view assigned tasks at `/my_tasks`, updating status (pending, in\_progress, completed) and comments.
  - Admins view all tasks and officer updates at `/transcripts`.
- **Hindi Support:** UTF-8 encoding and Noto Sans Devanagari ensure proper rendering of designations and task data.
- **Debugging:** `/debug_officers` endpoint for verifying `officers.json` loading.

### 5 Development Process

The project was developed iteratively, with key milestones:

1. **Setup:** Initialized Flask app, integrated Supabase, and configured `officers.json` loading.
2. **Authentication:** Implemented `/signup` and `/login` with Supabase Auth, later disabling email verification.
3. **Task Management:** Developed `/save_transcript`, `/get_transcripts`, and `/get_my_tasks` for task CRUD operations.
4. **Frontend:** Designed templates with Tailwind CSS and fixed Hindi text rendering issues.
5. **Debugging:** Addressed errors like `SyntaxError` in `transcripts.html`, module errors in `app.py`, and task update validation.
6. **Deployment:** Deployed on Render with environment variables for Supabase and Flask.

## 6 Challenges and Resolutions

1. **Challenge:** Login error “Email not confirmed” blocking user access.
  - **Resolution:** Disabled email verification in Supabase dashboard and added auto-login in `/signup` (June 9, 2025).
  - **Impact:** Immediate login post-sign-up, improving user experience for testing.
2. **Challenge:** `SyntaxError` in `transcripts.html` due to JSON parsing of `officers.json`.
  - **Resolution:** Used `{{ officers | tojson | safe }}` in templates and escaped quotes in `load_offices` (June 9, 2025).
  - **Impact:** Correct rendering of officer designations in dropdowns.
3. **Challenge:** Error “‘module’ object is not callable” in `/get_my_tasks` for officers.
  - **Resolution:** Fixed typo `json(()) . jsonify(())` to proper Supabase query (June 9, 2025).
  - **Impact:** Officers can view assigned tasks at `/my_tasks`.
4. **Challenge:** Error “`TypeError: 'NoneType' object is not iterable`” when officers update tasks.
  - **Resolution:** Modified `/update_task` to validate against `task_assignments` and restricted officer-select to assigned designations in `my_tasks.html` (June 9, 2025).
  - **Impact:** Officers can update status and comments, visible to admins at `/transcripts`.
5. **Challenge:** Tailwind CSS CDN warning for development builds.

- **Resolution:** Recommended Tailwind CLI setup for production; retained CDN for testing (June 9, 2025).
- **Impact:** Temporary solution for development; CLI setup pending for optimized builds.

## 7 Current Status

As of June 9, 2025, the application is functional with the following status:

- **Authentication:** Email verification disabled, enabling immediate login post-sign-up.
- **Task Management:** Admins can create and view tasks; officers can view and update assigned tasks, with updates visible to admins.
- **Frontend:** Templates render correctly with Hindi text, styled via Tailwind CSS CDN.
- **Backend:** All endpoints (/save\_transcript, /get\_transcripts, /get\_my\_tasks, /update\_task, /update\_transcript) are operational.
- **Database:** Supabase tables store tasks, assignments, roles, and designations; Row-Level Security (RLS) disabled for testing.
- **Deployment:** Live on Render at <https://nic-speech-summarizer.onrender.com>.
- **Pending Tasks:**
  - Implement Tailwind CLI for production builds.
  - Re-enable RLS with policies for secure data access.
  - Consider re-enabling email verification for production with a custom email provider.

## 8 Future Enhancements

- **Speech-to-Text Integration:** Add real-time Hindi speech transcription using APIs like Google Speech-to-Text.
- **Notifications:** Implement email or in-app notifications for task assignments and updates.
- **Search and Filters:** Add search functionality for tasks by officer, status, or deadline.
- **Mobile Optimization:** Enhance responsive design for mobile users.
- **Security:** Re-enable RLS and email verification; add CSRF protection.

## 9 Conclusion

The Hindi Speech Transcription and Task Management Application successfully meets its core objectives, providing a platform for task creation, assignment, and updates with Hindi-language support. Despite challenges like authentication errors, frontend parsing issues, and backend validation bugs, all were resolved through iterative debugging and code enhancements. The application is deployed and functional, with opportunities for future improvements in speech integration, security, and user experience.