CSCI 598 – Advanced Topics in Computer Science

Project: Buddy – Web-Based Real-Time Note-Taking Service

Aditya Jekkula (012609130)   Ibaad Mohammed (012568102)

Project Overview

Buddy is a server-side web application built with Django, Python, and Bootstrap. Users log in via their browser, start/stop audio recordings, watch live transcriptions in real time, and—when finished—see AI-generated summaries of their conversations. Everything runs on your Django server: no separate mobile or React app needed.

Technology Stack

- **Backend:** Django, Django REST Framework, Django Channels (WebSockets), Celery (async tasks)
- **Frontend:** Django templates + Bootstrap 5, plain JavaScript for `MediaRecorder` & WebSocket client
- **Speech-to-Text:** OpenAI Whisper or Google Cloud Speech-to-Text API
- **Summarization:** OpenAI GPT API (or Hugging Face Transformers)
- **Storage:** PostgreSQL for metadata & transcripts, AWS S3 (via `django-storages`) for audio files
- **Authentication:** Django's built-in auth system

Core Functionality

1. **User Authentication**—Sign up, log in/out, password reset via Django's auth and Bootstrap-styled forms.
2. **Live Audio Capture & Streaming**—Browser uses HTML5 `MediaRecorder` to capture audio chunks. Chunks are sent over WebSocket (Django Channels).
3. **Real-Time Transcription**—Server-side consumer buffers audio frames, processes with Whisper (or Google STT), and returns interim transcripts.
4. **Asynchronous Summarization**—On recording stop, a Celery task concatenates the transcript and calls GPT API for a bullet-point summary saved in the database.
5. **Notes Management**—Browse, search, edit, delete recordings and summaries; export notes as PDF or email.

Key Features

- Real-Time Speech-to-Text via Django Channels
- Live Transcript Display styled with Bootstrap
- AI-Powered Summaries queued through Celery

- Persistent Storage of audio (S3) and text (PostgreSQL)
- Server-rendered Bootstrap UI templates
- Secure Authentication with per-user isolation
- Export: PDF generation & email delivery

UI Pages & Navigation

- `/accounts/login/`: User login
- `/accounts/signup/`: User registration
- `/`: Dashboard (recent recordings & summaries)
- `/record/`: Recording page with live transcript
- `/notes/`: Notes list (search, edit, delete)
- `/notes/{id}/`: Detail view (transcript, summary, export)
- `/settings/`: Profile settings & logout

Next Steps

1. Define Django models: `Recording`, `TranscriptChunk`, `Summary`.
2. Implement Channels consumer for streaming STT.
3. Scaffold Celery tasks for final transcription & summarization.
4. Create Django views/templates for recording and notes.
5. Configure AWS S3 storage and static files.