Major Project - II
CS- 805

Project Title: Ai Mock Interview

Problem Statement: Job seekers lack personalized interview preparation. Our AI-powered platform generates domain-specific questions, transcribes voice responses, analyzes answers, and provides real-time feedback to improve interview readiness.

Project Group Number: 36

Group Members Details: Shubham Yadav (0187CS223D03)
Saurabh Vishwakarma (0187CS211154)
Vishal Sisodiya (0187CS211184)
Satyam Rajak (0187CS211151)

Project Guide: Prof. Amit Rathore

Idea/Approach Details

Describe your idea Solution/Prototype :

AI-powered mock interview web application serves as an intelligent interview preparation tool. It personalizes mock interviews by generating relevant questions based on the user's experience, domain, and technology. Users answer via voice, which is transcribed into text and analyzed by AI to provide instant feedback and suggest improvements. This solution enhances interview readiness by offering a structured, AI-driven evaluation process.

Abstract:

AI-Powered Mock Interview Web App helps users prepare for job interviews by generating personalized questions based on their experience, domain, and technology. Users answer via voice, which is converted to text and analyzed by AI to provide feedback and suggested improvements. Using the Gemini API, the platform enhances interview readiness with real-time evaluation and insights.

Describe your Technology stack :

- **Frontend**: Next.js.
- **Backend:** JavaScript, Nodejs.
- **Database**: PostgreSQL.
- **Real Time Monitoring:** WebSocket
- **Deployment:** Docker.

Project Requirements

Client-Side Hardware Requirements:

- **Device**: Smartphone, Desktop, Tablet.
- **Internet Connection**: Stable and fast connection
- **RAM**: At least 4GB RAM
- **Processo**r: Minimum Intel i3 / AMD Ryzen 3 or higher.

Client-Side Software Requirements:

- **Operating System:** Windows or Android
- **Browser**: Any latest version browser.

Developer-Side Hardware Requirements:

- **Device**: Desktop or laptop.
- **Memory (RAM)**: 4GB or more.
- **Internet Connectio**n: Stable and fast connection

Developer-Side Software Requirements:

- **Operating System**: Windows
- **IDE**: VS Code.
- **Browser**: Any latest version browser.
- **Backend Framewor**k: Next.js.
- **Database Management**: PostgreSQL.
- **Version Control**: Git & GitHub for code management

Project Requirements

Functional Requirements:

- User Authentication & Profile Management
- Interview Creation
- Voice Input & Text Conversion
- AI Response Analysis & Feedback
- Interview History & Insights
- Responsive & Interactive UI
- Database & Data Storage

Non-Functional Requirements

- Performance Scalability
- Reliability
- Security
- Usability
- Availability
- Compatibility
- Maintainability

Design

Use Cases:

Actors:

1.User – A person who log in and uses the platform to practice interviews.

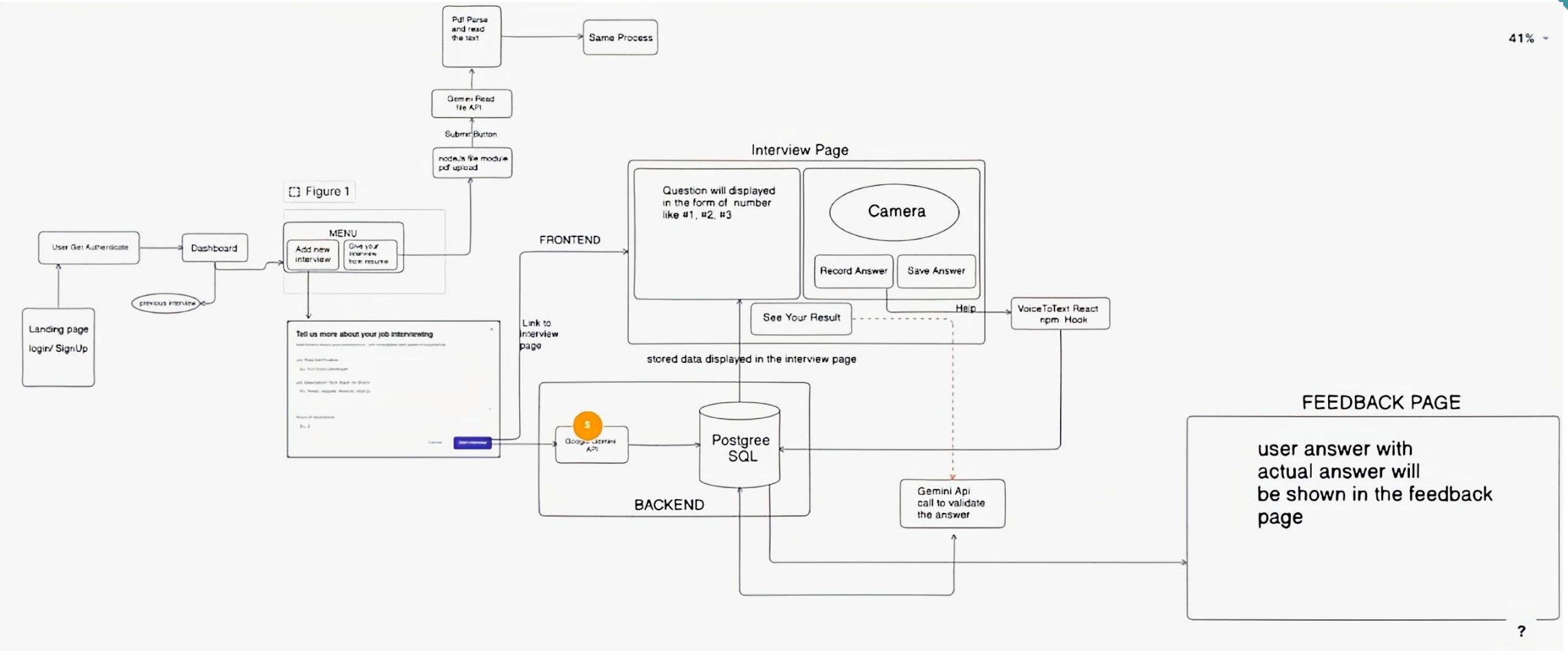
2.System(Ai Engine)– Processes input, generates questions, converts speech to text, analyzes responses, and provides feedback.

Primary Use Cases:

1. **User Login & Authentication -** User registers or logs in using Clerk authentication. System verifies credentials and grants access.
2. **Creating an Interview** - User inputs years of experience, work profile domain, and technology. System generates interview questions using the Gemini API.
3. **Answering Interview Questions -** System displays a question. User provides an answer via voice. System converts speech to text.
4. **AI Response Analysis & Feedback -** System analyzes the user's answer. Provides feedback on correctness and suggests improvements.
5. **Viewing Interview History & Insights** - User accesses past interviews and performance analytics.

Design

Diagram:



Deployment Details

Describe Deployment Details here:

Hardware Deployment:

•**System architecture** – Is it a web-based system, mobile app, or desktop software

•**Hosting environment** – It will be deployed on a cloud provider

•**Technology stack** – Frontend: HTML, CSS, JavaScript, Next
Backend: JavaScript, Nodejs
Database: PostgreSQL
RealTime Update: WebSocket
Version Control : GitHub

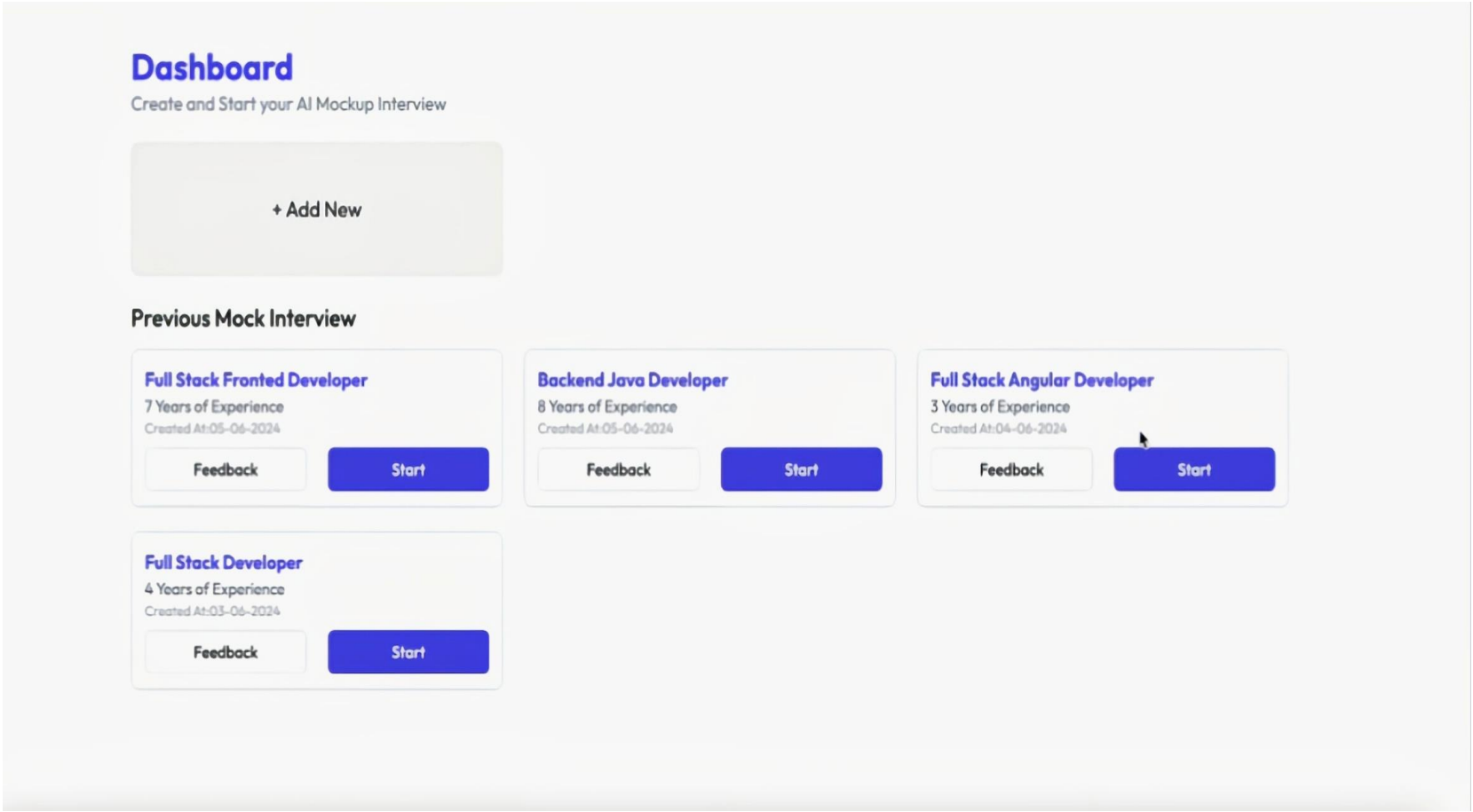
Maintenance

•**Preventive Maintenance**:
•Regular system audits.
•Update security patches and libraries.

•**Corrective Maintenance**:
•Identify and fix bugs through monitoring (e.g., **Prometheus**, **Grafana**).

•**Adaptive Maintenance**:
•Ensure compatibility with new hardware or software changes

Project Screenshots




Thank you!