

Project Documentation

1 Introduction

- **Project Title:** *EduTutorAi: Personalized Learning Platform*
- **Team Lead :** Ponduri Venkata Sai Lakshmi Deepthi
- **Team Members:**
 - **Palla Rajesh** — System Architect & Database Engineer
 - **Parasa Vennela Kranthi** — Backend Developer
 - **Paidimarri Venkata Sai Srilatha** — Full Stack Developer
 - **Ponduri Venkata Sai Lakshmi Deepthi** — AI/ML Engineer

2 Project Overview

- **Purpose:**

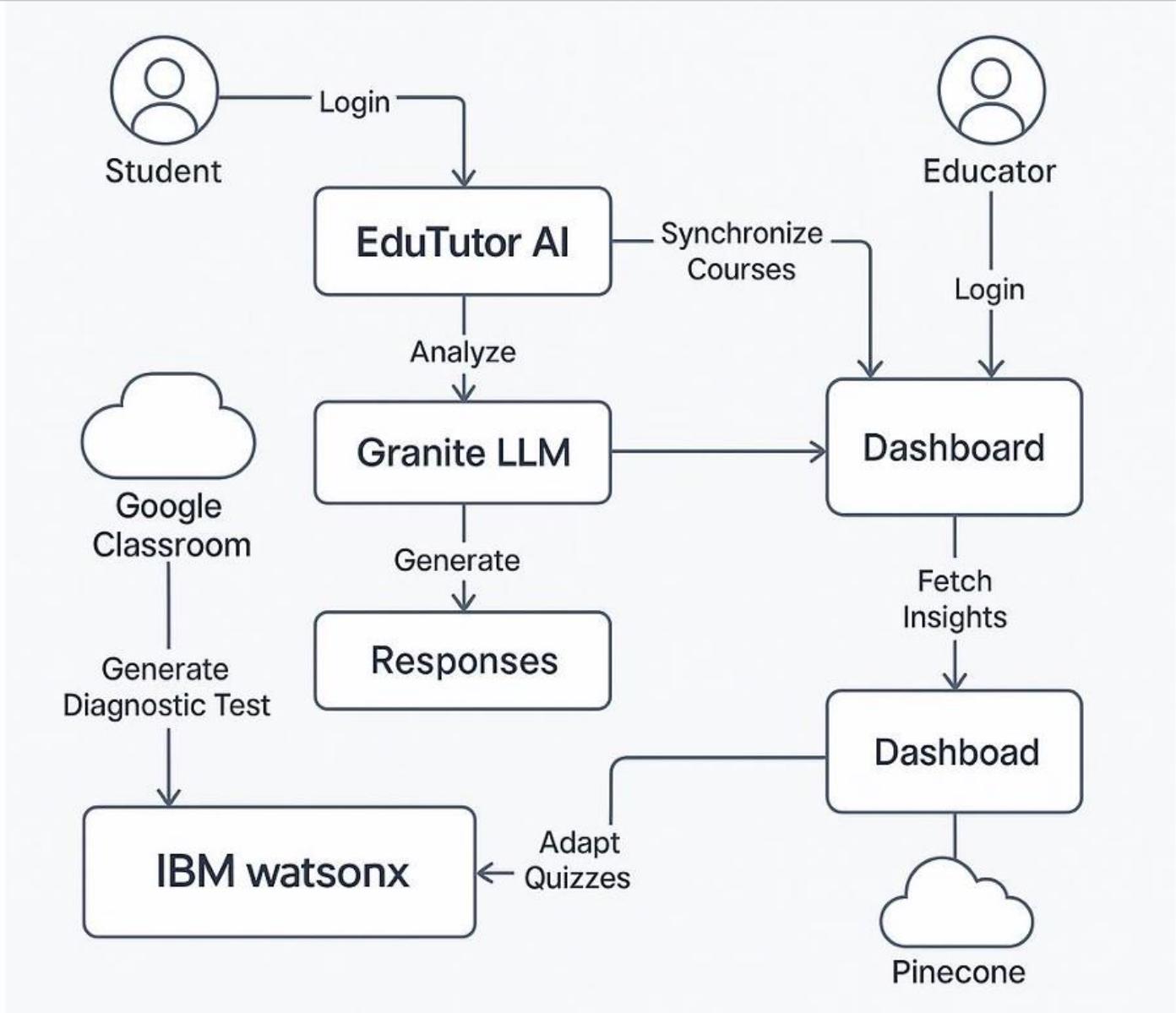
EduTutorAi is an AI-driven, personalized education platform built on the FastApi. It empowers students with dynamically generated quizzes, tracks their performance, and allows educators to view student analytics in real-time. Uses Google classroom.
- **Key Features:**
 - Google OAuth authentication
 - Google Classroom course sync
 - Dynamic quiz generation via AI
 - Real-time educator dashboard
 - Adaptive quiz difficulty
 - Secure student and educator login
 - Performance tracking and feedback

3 Architecture

- **Frontend (React):**
 - React functional components with hooks
 - React Router for client-side routing
 - Context API for global state management (authentication and user data)
 - Axios for API communication
- **Backend (Node.js + Express.js):**
 - RESTful APIs built using Express.js
 - JSON Web Token (JWT) for authentication
 - Routes for users, quizzes, and admin analytics
 - AI integration API endpoints for quiz generation

- **Database (MongoDB):**

- User collection (students & educators)
- Quiz collection (questions, answers, difficulty level)
- Results collection (student scores, feedback, date, topic)
- Diagnostic collection for initial skill level assessment



4 Setup Instructions

- **Prerequisites:**

- FastApi
- Pinecone vector DB
- Git

Installation & Setup Guide for EduTutor AI

1. Clone the Repository

Open your terminal and run:

```
bash
CopyEdit
git clone https://github.com/your-username/edututor-ai.git
cd edututor-ai
```

2. Backend Setup

a. Navigate to the backend folder

```
bash
CopyEdit
cd server
```

b. Create and activate a Python virtual environment (optional but recommended)

```
bash
CopyEdit
python -m venv venv
# Windows
venv\Scripts\activate
# macOS/Linux
source venv/bin/activate
```

c. Install backend dependencies

```
bash
CopyEdit
pip install -r requirements.txt
```

3. Frontend Setup

a. Navigate to the frontend folder

```
bash  
CopyEdit  
cd ../client
```

b. Install frontend dependencies

```
bash  
CopyEdit  
npm install
```

4. Environment Variables Setup

a. Backend environment variables

Create a .env file inside the server directory with your configuration, e.g.:

```
ini  
CopyEdit  
DATABASE_URL=your_database_url  
PINECONE_API_KEY=your_pinecone_api_key  
PINECONE_ENVIRONMENT=your_pinecone_env  
GOOGLE_CLIENT_ID=your_google_client_id  
GOOGLE_CLIENT_SECRET=your_google_client_secret  
SECRET_KEY=your_secret_key
```

Make sure to replace placeholders with your actual keys and secrets.

5. Run the Project

a. Start the backend server

From the server directory, run:

```
bash
```

```
CopyEdit  
uvicorn main:app --reload
```

b. Start the frontend development server

From the client directory, run:

```
bash  
CopyEdit  
npm start
```

6. Access the Application

Open your browser and navigate to:

```
arduino  
CopyEdit  
http://localhost:3000
```

or wherever your frontend server is configured.

5 Folder Structure

- Client

```
└── templates/
    ├── base.html
    ├── login_role_select.html
    ├── student_dashboard.html
    ├── educator_dashboard.html
    ├── quiz_generate_form.html
    ├── quiz_questions.html
    ├── quiz_result.html
    └── diagnostic_test.html
└── static/
    └── style.css
└── main.py
```

- Server

```
─ app/
  └── api/
      ├── routes/
      │   ├── auth.py
      │   ├── quiz.py
      │   ├── student.py
      │   └── educator.py
      └── __init__.py
  └── core/
      ├── config.py
      ├── oauth.py
      ├── granite_client.py
      └── pinecone_client.py
```

6 Running the Application

Frontend

- **Navigate to the client folder:**

```
bash  
CopyEdit  
cd client
```

- **Install frontend dependencies:**

```
bash  
CopyEdit  
npm install
```

- **Start the frontend server:**

```
bash  
CopyEdit  
npm start
```

Backend

- **Navigate to the project root:**

```
bash  
CopyEdit  
cd EdututorAI
```

- **(Optional) Create a virtual environment:**

```
bash  
CopyEdit  
python -m venv venv  
source venv/bin/activate # Windows: venv\Scripts\activate
```

- **Install backend dependencies:**

```
bash  
CopyEdit  
pip install -r requirements.txt
```

- **Ensure environment variables are configured:**
- Create a .env file in the root:

```
env
CopyEdit
OPENAI_API_KEY=your_openai_key
PINECONE_API_KEY=your_pinecone_key
PINECONE_ENV=your_pinecone_env
GOOGLE_CLIENT_ID=your_google_client_id
GOOGLE_CLIENT_SECRET=your_google_client_secret
```

- Place the client_secret.json in the root directory for Google OAuth integration.
- **Run the FastAPI server:**

```
bash
CopyEdit
uvicorn app.main:app --reload
```

7 API Documentation

Example API Endpoints:

- **GET /login**

Description: Initiates OAuth login with Google.

Response: Redirects to Google OAuth consent screen.

- **GET /auth/callback**

Description: Handles Google OAuth callback and stores user info in session.

Response: Redirects to dashboard based on user role (student or educator).

- **GET /Logout**

Description: Logs the user out and clears session.

Response: Redirects to login page.

GET /student/quiz

- **Description:** Shows the quiz generation form for students.
- **Response:** Renders quiz_generate_form.html.

POST /student/generate_quiz

- **Description:** Generates quiz questions using Granite AI model.

POST /student/submit_quiz

- **Description:** Submits quiz responses, evaluates score, stores in Pinecone.
- **Form Data:** User's selected answers + hidden fields (topic, difficulty)
- **Response:** Renders `quiz_result.html`.

GET /student/diagnostic

- **Description:** Displays diagnostic test results after quiz submission.
- **Response:** Renders `diagnostic_test.html`.

GET /educator/dashboard

- **Description:** Displays overview of student quiz results for educators.
- **Response:** Renders `educator_dashboard.html`

8 Authentication

- **Mechanism:** JWT (JSON Web Tokens)
- **Storage:** Token stored in browser localStorage
- **Workflow:**
 - User logs in via Google OAuth or custom form
 - Backend verifies and issues JWT
 - Frontend sends token in headers for secured routes

9 User Interface

Sample Screens:

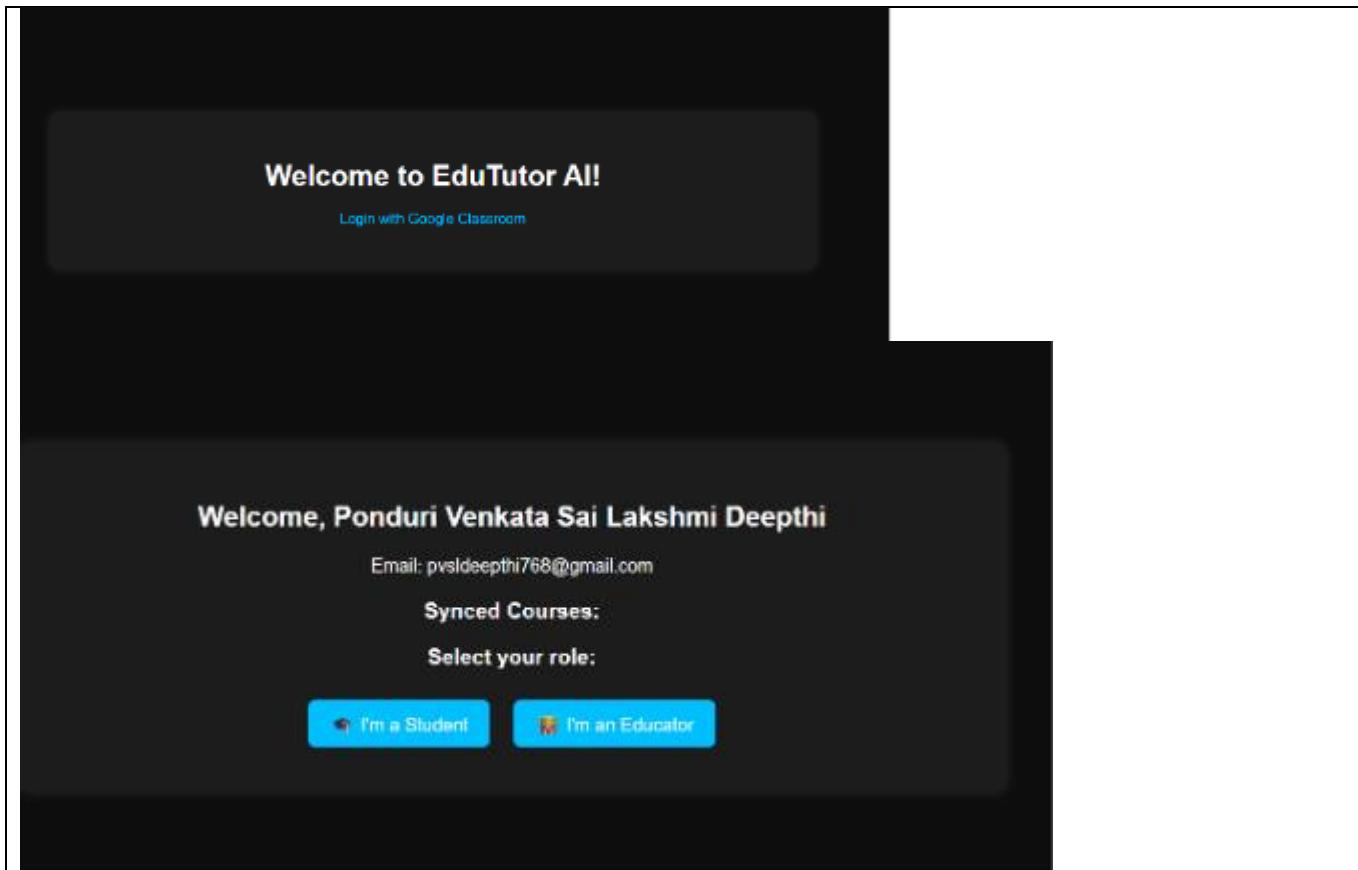
- Login/Register screen
- Student Dashboard
- Quiz Attempt Page
- Results & Feedback Page
- Educator Performance Dashboard

Tools: React with Material-UI / Tailwind CSS

10 Testing

- **Strategy:**
 - Unit Testing (Jest, Mocha) for backend APIs
 - Component testing (React Testing Library) for frontend
 - End-to-end testing using Cypress

1 1 Screenshots or Demo



Sign in with Google

You're signing back in to EduTutor Ai

Review EduTutor Ai's Privacy Policy and Terms of Service to understand how EduTutor Ai will process and protect your data.

To make changes at any time, go to your [Google Account](#).

Learn more about [Sign in with Google](#).

pvslakshmi76@gmail.com

Cancel Continue

Sign in with Google

Choose an account

To continue to [EduTutor Ai](#)

- pvslakshmi76@gmail.com
- Solver [programmer76@gmail.com](#)
- Shak runessa [runessa10@gmail.com](#)
- Tech [tech1000@gmail.com](#)
- Shak Runessa [runessa100@gmail.com](#)
- Use another account

Welcome, Ponduri Venkata Sai Lakshmi Deepthi

Your Courses:

mathematics 2nd btech

Select subject for Diagnostic Test:

mathematics 2nd btech

Start Diagnostic Test

Enter topic for Quiz:

probability

Difficulty:

Medium

Generate Quiz

Logout

Welcome, Ponduri Venkata Sai Lakshmi Deepthi

Your Courses:

mathematics 2nd btech

Select subject for Diagnostic Test:

mathematics 2nd btech

[Start Diagnostic Test](#)

Enter topic for Quiz:

Enter topic

Difficulty:

Medium

[Generate Quiz](#)

[Logout](#)

Welcome, Ponduri Venkata Sai Lakshmi Deepthi

Your Courses:

mathematics 2nd btech

Select subject for Diagnostic Test:

mathematics 2nd btech

[Start Diagnostic Test](#)

Enter topic for Quiz:

Enter topic

Difficulty:

Medium

[Generate Quiz](#)

[Logout](#)

Quiz Results - probability

Score: 0/6

[← Back to Dashboard](#)

Diagnostic Test

[Submit Diagnostic Test](#)

You have been successfully logged out.



[Go back to login again.](#)

1 **2** Known Issues

- Quiz questions occasionally repeat due to limited AI data pool
- No offline mode (requires constant internet)
- Slight delay in AI-based quiz generation

1 **3** Future Enhancements

- Integrating video-based learning recommendations
- Advanced AI-based adaptive quizzes with personality mapping
- Mobile app (React Native)
- Offline quiz attempt support
- Gamified leaderboards