

# EduAssist | Team 38

Software Engineering Project - IIT Madras BS



# Agenda

- 1 Meet our team
- 2 Project overview
- 3 User segmentation
- 4 Problem statements
- 5 Design & feature implementation



# Meet Our Team



Rishav Kumar  
Project manager



Treasa Janet  
Scrum manager



Palash Johri  
Frontend & Code manager



Sohini Sarkar  
Backend



Prabhu  
Backend & Testing



Raghavendra Narayan Jha  
Frontend



Omkar Shankar Pawar  
Backend



# Project Overview & Users

## Primary Users

### IITM BS Teaching Assistants

They are responsible for managing day-to-day course operations such as grading assignments, monitoring student progress, answering queries, and assisting instructors in content delivery.

They rely on the system for streamlined communication, efficient task management, and easy access to course data.

## Secondary Users

### IITM BS Support Team

They use the application to track course performance, allocate teaching resources, and ensure compliance with institutional policies. Their focus is on maintaining smooth program operations and ensuring data consistency across courses.

## Tertiary Users

### IITM BS Course Instructors

Course instructors are not the main users of the system, they occasionally interact with it to upload materials, review student performance summaries, and coordinate with TAs and administrators. Their goal is to ensure teaching quality and learning effectiveness.

# Problem Statements

## Addressing Repetitive queries

Needs to answer same/similar queries again and again for the students.

## No proper segmentation of teaching content

Student usually seek for topic wise content which is not provided every time.

## Burden of administering tasks

TAs usually have a lot of administrative tasks other than their core job of being a teaching assistant.

## Time consuming teaching content generation

For every session with the students, TAs need to generate some teaching content which is a tedious task.



# Proposed Solution: Summary of Features



## RAG Chatbot

For answering repetitive queries



## Content Tagger

For segmentation of teaching content



## Workflow Agent

Automating the tasks of TAs



## Assessment Generator

For constant creation of assessments  
as per the coursework



## Slide Deck Generator

For easy creation of presentation for  
teaching

# Design and Implementation - Wireframe

The dashboard features a sidebar with 'EduAssist AI Teaching Assistant' and a main area titled 'EduAssist Dashboard'. It includes sections for 'AI Tools' (Knowledge Assistant, Study Guide Generator, Admin Workflow Agent, Assessment Generator, Slide Deck Creator) and 'Analytics' (Time Saved: 25 hours, +10%).

**EduAssist**  
AI Teaching Assistant

**EduAssist**

Features   Analytics   Settings

**Dashboard**

**AI Tools**

**Knowledge Assistant**  
Get instant answers to student queries.

**Study Guide Generator**  
Generate comprehensive study guides.

**Admin Workflow Agent**  
Automate administrative tasks.

**Assessment Generator**  
Create assessments quickly.

**Slide Deck Creator**  
Design engaging slide decks.

**Analytics**

Time Saved  
**25 hours**  
+10%

The sign-in page has fields for 'Email' and 'Password', a 'Login' button, and links for 'Take me to Sign up' and 'Forgot Password?'. It includes a background illustration of a person at a desk.

**EduAssist**

**Sign in to your account**

Email  
Enter your email

Password  
Enter your password

Login

Take me to Sign up

Forgot Password?

The account creation page has fields for 'Name', 'Email', 'Password', 'Course Code', 'TA Role Verification', and a 'Create Account' button. It includes a background illustration of a person working on a laptop.

**EduAssist**

**Create your account**

Name  
Enter your name

Email  
Enter your email

Password  
Enter your password

Course Code  
Enter your course code

TA Role Verification  
Enter your role

Create Account

Already have an account? Login

# Knowledge Assistant

EduAssist

Features Analytics John Doe

## AI Knowledge Assistant

Search student questions

### Recent Questions

What is quantum entanglement? 2 hours ago

The concept of 'quantum entanglement' refers to a physical phenomenon where two or more particles become linked in such a way that they share the same fate, even when separated by large distances. This means that the quantum state of each particle cannot be described independently of the state of the others, even if the particles are far apart. When you measure a property of one particle, you instantly know the corresponding property of the other particle, regardless of the distance between them. This is because the particles are entangled, and their fates are intertwined. This concept is fundamental to quantum mechanics and has been experimentally verified. It's important to note that entanglement does not allow for faster-than-light communication, as the measurement outcome on one particle is random and cannot be controlled to send a specific message.

Confidence: High

Explain the Central Limit Theorem. 4 hours ago

The 'Central Limit Theorem' (CLT) is a fundamental concept in probability theory and statistics. It states that the distribution of the sum (or average) of a large number of independent, identically distributed random variables will be approximately normally distributed, regardless of the original distribution of the variables. This holds true as long as the random variables have a finite variance. In simpler terms, if you repeatedly sample from any population (with a finite variance) and calculate the mean of each sample, the distribution of these sample means will tend towards a normal distribution as the sample size increases. The CLT is crucial because it allows us to make inferences about population parameters (like the mean) even when we don't know the exact distribution of the population, as long as we have a sufficiently large sample size.

Confidence: Medium

What is the Traveling Salesman Problem? 6 hours ago

The 'Traveling Salesman Problem' (TSP) is a classic optimization problem in computer science and operations research. It asks the following question: Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city? This problem is NP-hard, meaning that there is no known efficient algorithm to find the optimal solution for large instances of the problem. However, there are various heuristic and approximation

### AI Response

The concept of 'quantum entanglement' refers to a physical phenomenon where two or more particles become linked in such a way that they share the same fate, even when separated by large distances. This means that the quantum state of each particle cannot be described independently of the state of the others, even if the particles are far apart. When you measure a property of one particle, you instantly know the corresponding property of the other particle, regardless of the distance between them. This is because the particles are entangled, and their fates are intertwined. This concept is fundamental to quantum mechanics and has been experimentally verified. It's important to note that entanglement does not allow for faster-than-light communication, as the measurement outcome on one particle is random and cannot be controlled to send a specific message.

### Citations

Quantum Mechanics: Concept...  
Experimental Verification of Qu...

"As a Teaching Assistant, I want a feature where a discussion forum automatically suggests similar questions that have already been answered as a student type of their new question"



# Content Priority Tagger

● EduAssist

EduAssist  
AI Teaching Assistant

- Dashboard
- Knowledge Assistant
- Study Guide Generator
- Admin Workflow Agent
- Assessment Generator
- Slide Deck Creator

Content Priority Tagger  
Tag and organize lecture content by priority.

Tag Content

Collect Feedback

## Content Priority Tagger

Upload Lecture Video  
Drag and drop or browse

Upload

### Timeline with Priority Tags

- Introduction to AI  
High Priority
- Machine Learning Basics  
Medium Priority
- Deep Learning Concepts  
High Priority
- Advanced AI Techniques  
Low Priority
- Conclusion and Q&A  
Medium Priority

### Generated Study Guide

Based on the lecture video and priority tags, a study guide has been generated to help students focus on key concepts and areas of high importance. This guide includes summaries, key terms, and practice questions.

"As a Teaching Assistant, I want some feature like where I can tag course materials like topics, questions, or video timestamps as 'High-Priority'"

An illustration depicting a classroom environment. In the foreground, a student is seated at a desk, looking at a tablet device. Another student is standing behind them, writing on a whiteboard. The room is filled with educational equipment like a computer monitor and a projector screen displaying graphs. The overall theme is modern education and technology integration.

# Workflow Agent

◆ EduAssist

Features ▾ Analytics John Doe

EduAssist  
AI Teaching Assistant

Inbox Drafts Sent Archived Deleted

Dashboard Knowledge Assistant Study Guide Generator Admin Workflow Agent Assessment Generator Slide Deck Creator

## Extension Requests

Student: Alex Chen  
Assignment 2 - Deadline Extension 2d ago

### Request Summary

Alex Chen, a student in your class, has requested an extension for Assignment 2 due to a family emergency. They have attached supporting documentation. The student's current grade in the course is a B+, and they have not requested any extensions previously.

### AI Recommendation

Based on the student's academic history, the provided documentation, and the course extension policy, the AI recommends approving the extension request.

### Draft Email

Email Preview

Dear Alex, Thank you for reaching out regarding your request for an extension on Assignment 2. After reviewing your situation and the supporting documentation you provided, I am pleased to approve your extension request. You will have an additional 7

Deny Approve

<https://eduassist-nine.vercel.app/features/knowledge-assistant>

As a Teaching Assistant, I want a tool to help automate repetitive administrative tasks like tracking submissions etc"



# Slide Deck

EduAssist

AI Teaching Assistant

Dashboard

Knowledge Assistant

Study Guide Generator

Admin Workflow Agent

Assessment Generator

Slide Deck Creator

Slide Deck Generator  
Create professional slide decks from content.

Generate Slides

Collect Feedback

## Slide Deck Generator

Session Topic

Enter the session topic

Upload Notes

Upload notes to generate slides

Generate Outline

### AI-Generated Outline

Introduction to Machine Learning

Supervised Learning

Unsupervised Learning

Reinforcement Learning

Conclusion

### Slide Preview

Slide deck generator for creation of teaching presentation



# Assignment Generation

◆ EduAssist

EduAssist  
AI Teaching Assistant

Dashboard

Knowledge Assistant

Study Guide Generator

Admin Workflow Agent

Assessment Generator

Slide Deck Creator

Assessment Generator  
Create and manage assessments with AI assistance.

Generate Assessment

Collect Feedback

## Automated Assessment Generator

### Assessment Configuration

Select Topic

Difficulty Level

Number of Questions

Generate Questions

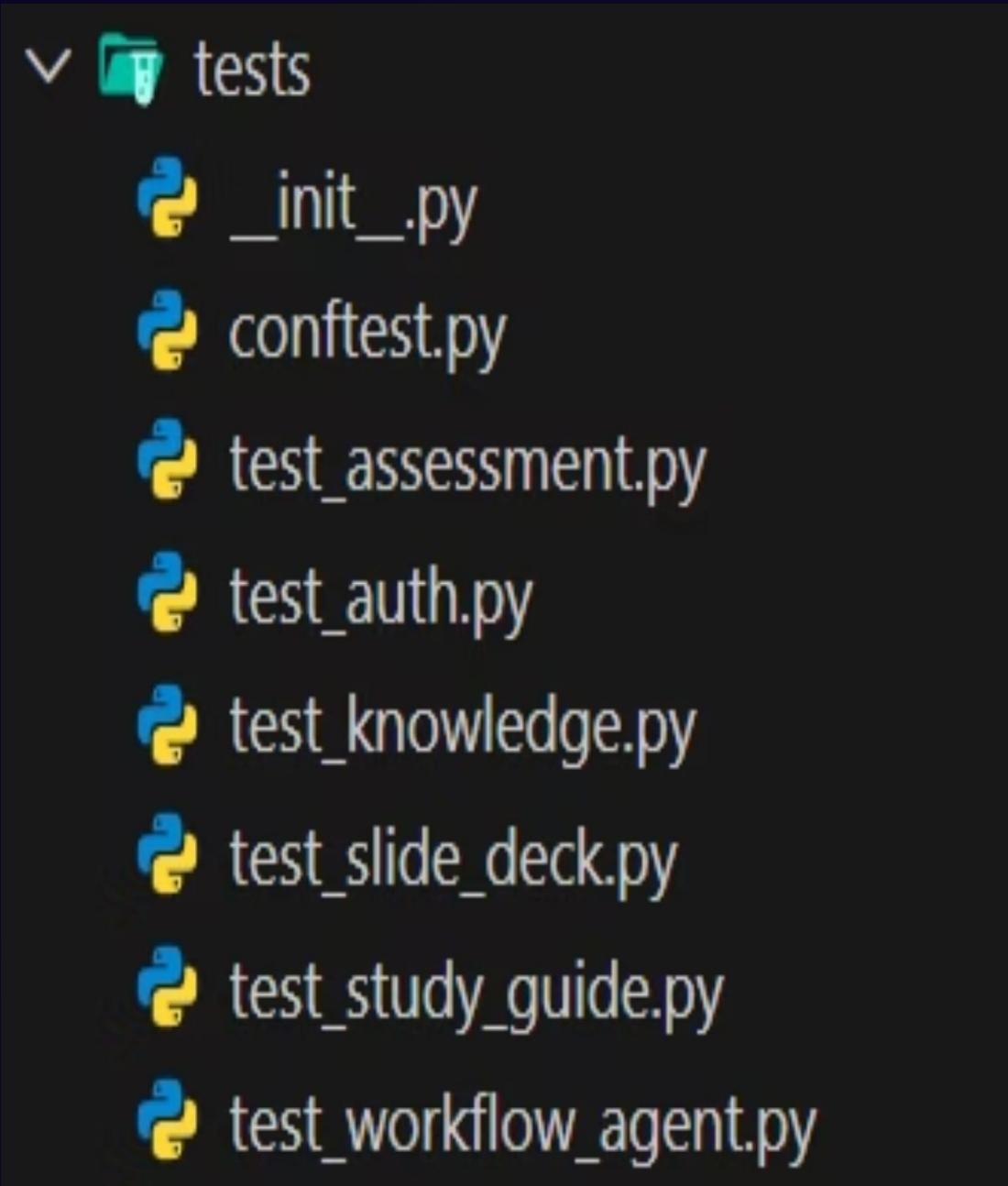
### Generated Questions

Question	Marks
Explain the concept of recursion with an example.	5
What are the differences between arrays and linked lists?	4
Describe the time complexity of binary search.	3
Implement a function to reverse a string.	6
Discuss the advantages and disadvantages of using a hash table.	4



Assignment generator for generating tests based on the coursework provided

# Testing & Validation



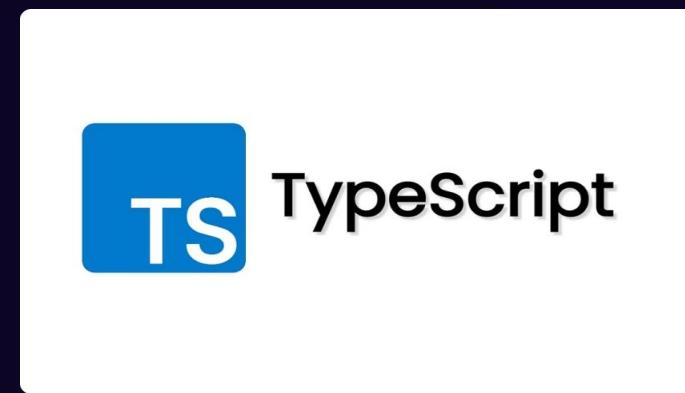
```
backend > tests > 🐍 test_auth.py > ...
"""
Tests for Authentication API
"""

import pytest
from httpx import AsyncClient

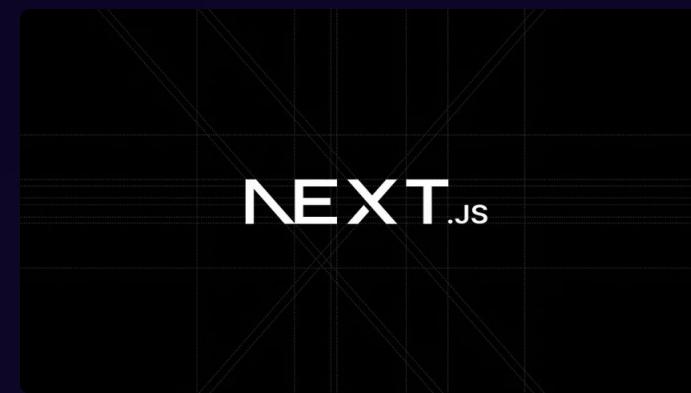

@pytest.mark.asyncio
async def test_register_user(client: AsyncClient):
    """Test user registration"""
    response = await client.post("/auth/register",
        json={
            "email": "newuser@example.com",
            "username": "newuser",
            "password": "password123",
            "full_name": "New User"
        }
    )
    assert response.status_code == 200
    data = response.json()
    assert data["username"] == "newuser"
    assert data["email"] == "newuser@example.com"


@pytest.mark.asyncio
async def test_login(client: AsyncClient, test_user):
    """Test user login"""
    response = await client.post("/auth/login",
        data={"username": "testuser", "password": "testpass123"}
    )
    assert response.status_code == 200
    data = response.json()
    assert "access_token" in data
    assert data["token_type"] == "bearer"
```

# Our Tech Stack



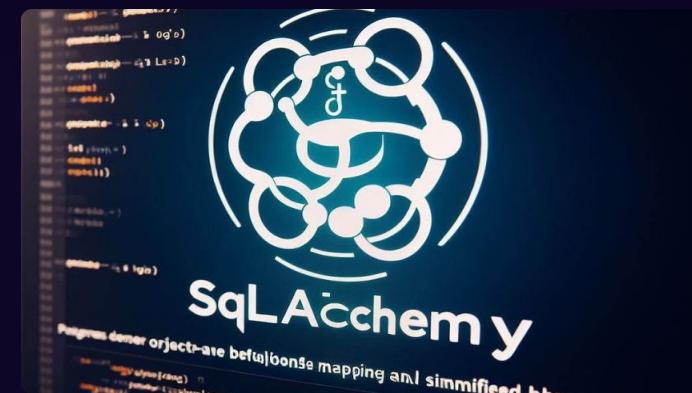
TypeScript



Next.js



Python



SQLAlchemy



Uvicorn ASGI Server



Gemini API



LangChain Framework



ChromaDB



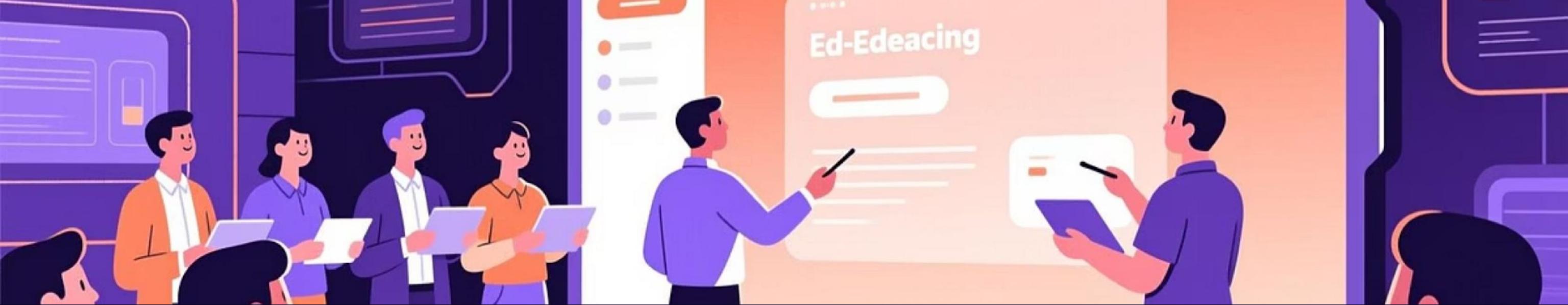
Clickup



GitHub

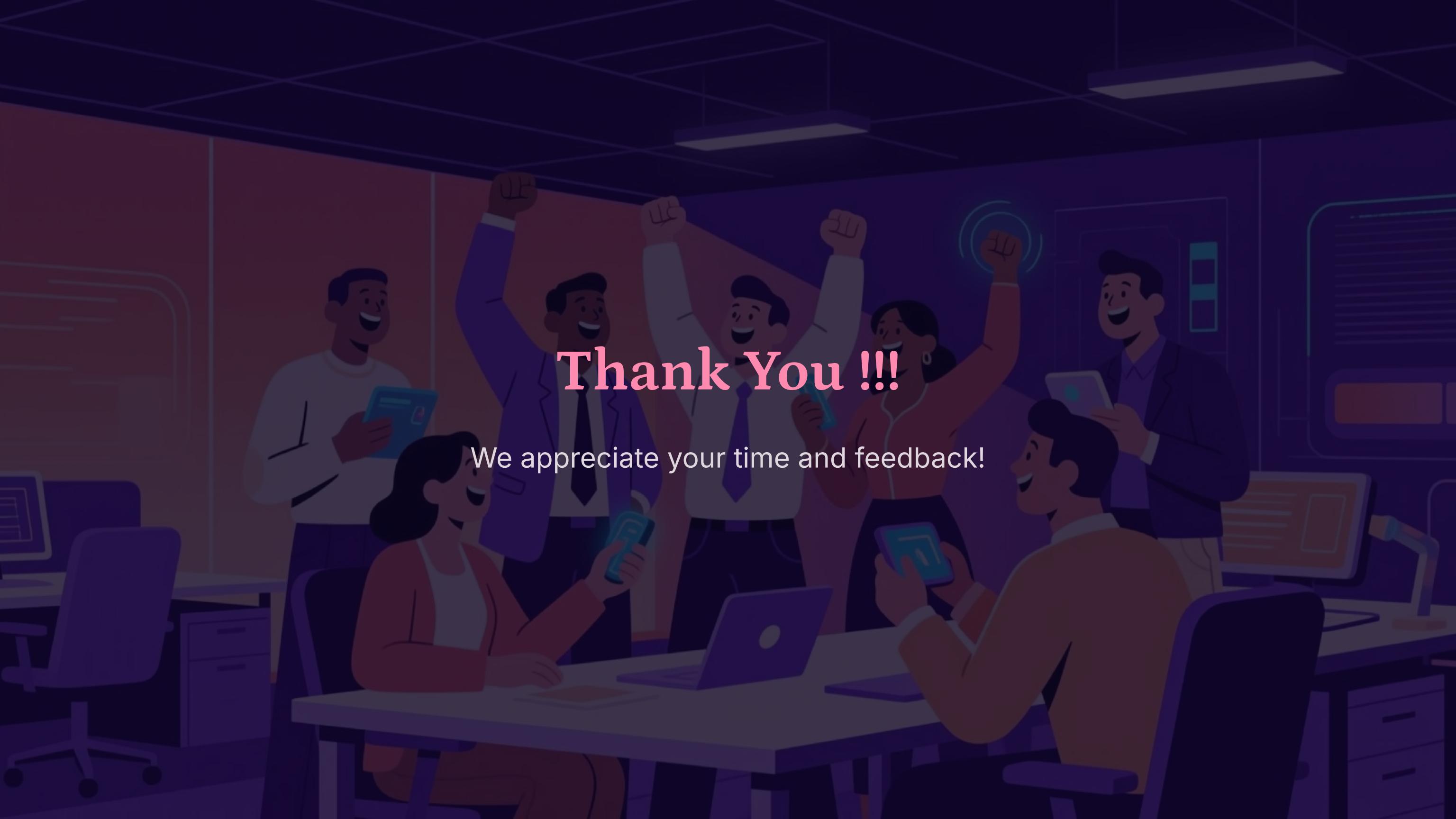


Figma



# Demo Time!

See our project in action



# Thank You !!!

We appreciate your time and feedback!