



Graduation Project

Smart Education

By

**Alzahraa Mohamed Shaheen
Marco Maher Anwar
Mohamed Said El-Seify
Eman Mohamed Gomaa
Mohamed Amir Mohamed**

**Bahaa Mohamed El-Said
Amir Hany Hussien
Mahmoud Mohamed Azab
Hossam El-Nabawy Ahmed
Abdelhamid Saber Abdelhamid**

Supervised by

Dr. Amr Zamel

Zagazig University
Faculty of Engineering
Computer and Systems Engineering

July 2022 © Copyright by **SmartEdu Team**

Acknowledgment

It has been a great opportunity to get a lot of real-world project experience as well as knowledge about how to design and analyze real projects. We would like to express our gratitude to Allah for enabling us to complete our project and reach this learning point.

We would like to express our deepest appreciation to Dr. Amr Zamel Ph.D. Professor, Computer and systems dep., for his supervision in harmony with us and guidance, continuous encouragement, and support during this project.

We are deeply grateful to all who supported us in all aspects.

Table of Contents

The preface	11
Significance and motivation	11
Aims and Objectives	12
Improving students' performance	12
Saving time and effort in the teaching process	12
Facilitating Communication during Courses	12
Methodology	13
Introduction:	14
Theoretical Background:	14
The previous studies and works	15
The courses that we have learned from:	16
Materials and Methods	17
Methods and Procedures:	17
Materials:	22
4.1 FrontEnd	27
4.1.1 Framework & programming language:	27
1- HTML 5:	27
2- CSS 3:	27
3- JavaScript :	28
4- jQuery:	28
5- BootStrap 5:	28
4.1.2. User Flow Diagram:	29
The user can be: Student or teacher.	29
Our goals:	29
For the teacher:	29
For the students:	29
First: Public home with Authentication	30
Second: Home for Students	31
Third: Home for Teacher	33

I locked the quiz :	36
4.2-Backend	39
4.2.1)Framework & programming language:	39
4.2.1.1)Why did we use Django web Framework?	39
4.2.1.2)Which design pattern does Django Follow?or What does django code look like?or What is Dango Framework process?	39
4.2.2)Database	42
introduction	42
overview of our Database & The ERD diagram:	42
The Business Roles:	42
The Entities & attributes:	46
The ER Diagram:	48
4.2.3)How did we use django to build the website?	49
Project Structure:	49
The Implementation of each application:	50
1-app_users:	50
2-classroom:	56
3-module:	66
4-assignment & schedule	69
5-quiz & schedule:	77
6-page & discussion	85
7-question & answer section	90
8-Chat Room(this feature will be improved in next version)	95
9-Leaderboard:	103
How did we build the API to be used by the mobile application ?	104
In short, what is API?	105
1-User Authentication:	106
1.1use:	106
1.2Sign In	107
1.3-Profile:	108
2-Classroom:	109
3-module:	110
4-page:	111
5-assignment:	112

6-quiz:	113
7-completion:	114
8-schedule:	115
ISSUES & CHALLENGES	115
Materials:	117
Category of Widgets:	118
Types of Widgets:	119
Model	122
View	122
Controller	123
Tools	123
Methods and Procedures:	126
Project Structure	126
Flutter State Management	127
What is State?	127
Stateless widget Vs Stateful widget.	127
Build UI from application state	128
What is Bloc?	128
API	133
Caching	136
Results and Discussion	148
Quiz Generator	150
MCQ Quiz	151
DL Model:	151
Transformers Theory	151
NLP	151
Why is it challenging?	152
Transformer models are used to solve all kinds of NLP tasks,	152
Transformers are language models	152
Transformers are big models	153
Transfer Learning	154

Architecture	155
Attention layer	155
Models Training	156
T5 Transformer	156
SQuAD (Stanford Question Answering Dataset)	157
RACE Dataset	158
SciQ Dataset	158
Training Results	158
Question Generation	158
Distractors Generation	159
Models Output	161
Question Generation	161
Distractors Generation	161
NLP Model:	162
Tokenize to Sentence using NLTK:	162
NLTK: Natural Language Toolkit	162
NLTK vs SpaCy	162
Implementation	163
Results	163
Extract Keywords (Answers) using PKE:	164
PKE	164
Theory	164
Implementation	165
Results	166
Find Distractors (Wrong Options) using Wordnet	166
Wordnet	166
Results	168
Filter Distractors using SciPy	168
SciPy	168
Theory	169
Implementation	169
Results	169
Final Output	170
Implementation	170
Results	170

T/F Quiz	170
False Statements	172
AllenNLP	173
Constituency Parsing	173
Result of Tree	174
Result of Terminate Noun Phrases or Verb Phrases Rightmost	174
Result of Complete Sentence using Transformer	174
Result of Filter Phrases using SciPy	175
Assignment Reviewer and OCR Reader	176
Tools and software:	176
Jupyter Lab (Notebook)	176
Google collaboratory (Colab)	176
OCR Reader	176
For reading pdf files using OCR part:	176
Methods and Modules:	177
Tesseract-ocr (1)	177
Pdf2image (2)	177
Poppler-utils (3)	178
Pillow (PIL) (4)	178
Clean-text (5)	178
cv2 (open cv) (6)	178
The steps to implement the task	179
Code implement :	180
1- Install requirements	180
2- Get pdf files	180
3- Convert pdf pages to images	180
4- Preparing the pictures to be read	180
5- Reading each image and get its text	181
6- Preparing the extracted text	181
7- The text is ready to use	181
Results	181

Notes	183
Assignment Reviewer	184
Grading Assignment	184
For grading programing assignment part:	184
Methods and Modules:	184
Subprocess (7) (Subprocess management) method	184
SequenceMatcher (8)	184
The steps to implement the task	188
Code implement :	188
1- Install requirements	188
2- Get the programming files of students and doctor	188
3- Call subprocess function and get the outputs	188
4- Get similarity ratio	189
Results	189
Notes	189
Plagiarism Detector	190
For plagiarism detector for programing assignments part:	190
Methods and Modules:	190
Scikit-learn (9)	190
TfidfVectorizer (10)	190
Cosine similarity	192
The steps to implement the task	193
Code implement :	194
1- Install requirements	194
2- Get the syntax of all programming files	194
3- Convert syntax to array and get the similarity between files	194
4- Get similar files according to the percentage Error! Bookmark not defined.	
Results	195
Notes	196

Chatbot:	196
AI Virtual Assistant:	196
Materials:	196
Watson Assistant: Intelligent virtual agent	196
Strength:	204
For Teacher:	204
For students:	205
Next Version	206
References	208

Abstract

After the global pandemic of Corona, studying from home is no longer just an option, the problem lies in the difficulty of communication between the teacher and students in a remote environment (for example, but not limited to: the difficulty of answering questions directly, the difficulty of scheduling the quizzes, answering questions in a system, etc...) as each of them has to use more than one communication platform. "Smart Edu" platform helps enhance communication between the teacher and his students, relying on a simplified user experience using the latest AI technology methods. Before starting work on the project from a technical point of view, we conducted user research to make sure that we use the best features and achieve inimitable simplicity in order to keep the platform in a simplified state.

After conducting the research and identifying the features, we measured the engineering possibilities of implementing the features and developed alternative features that achieve the same goal but at a lower cost. The final solution was to create an all-in-one platform that works on the web and mobile, that enables students to do all of their activities in one place. (e.g, view all educational content, communicate with the teacher, find out about upcoming exams and quizzes, monitor their grades, and most importantly communicate better with their teacher..etc, while the teacher can monitor the performance of all students, create new quizzes, send an announcement to all students...etc) all in one place.

Chapter 1

Introduction

The preface

Online learning is becoming more and more popular as technology develops and user experience gets better. When compared to traditional schooling, it has been shown to be a successful technique of learning and offers a variety of advantages. It has advantages, but it also has some challenges.

We collected most of those challenges to solve them in our online educational system to allow effective interaction between the professor and the student through many features that depend heavily on the advantages of artificial intelligence applications. We focused on both improving students' performance and facilitating professors' tasks through web and mobile applications in our project.

Significance and motivation

Since the beginning of the university's approach to distance learning, and with the increase in the number of platforms used, a large number of students faced the following problems:

1. Students have become more prone to distraction, so instead of focusing on improving their weaknesses, most focus on getting the most educational content from several resources for fear of missing any part.
2. Not all students know about projects or tasks that must be completed and delivered on a specific date
3. The student does not realise the real goal of his study of the scientific subject because he does not see what the professor of the subject seeks to convey.
4. The student does not realise his weaknesses in all scientific subjects, as there is no periodic evaluation in short periods, measuring his theoretical information or practical experience and providing customised solutions.
5. Difficulty in communication between the student and the professor or even his classmates

Aims and Objectives

→ Improving students' performance

1. Discussion on each topic for a full understanding of concepts and ease of access notes
2. Using points as an evaluation method
3. A leaderboard in each course to motivate students to collect more points and be at the top of the class
4. Students must complete each unit of the course before the date of accessing the next unit to get points
5. Students can collect points by watching videos before the deadlines, participating in discussions, submitting assignments and their grades, and also grades of solving quizzes.
6. Organising tasks by the deadlines schedule of units, assignments, and quizzes
7. Fast access to any material, or information by the chatbot
8. Locking quizzes when changing the tab to prevent cheating

→ Saving time and effort in the teaching process

1. Quizzes can be generated automatically from lectures files using AI
2. Programming assignments can be reviewed and detect cheating among students
3. Tracking students' progress through the leaderboard and their grades along the course
4. Replying to questions in discussions only once compared to private messages

→ Facilitating Communication during Courses

1. Video Meetings
2. Chat Rooms
3. Discussions by comments and replies on each topic

Methodology

We used the Scrum methodology, Scrum is a form of agile project management. work is split into short cycles known as sprints, which in our case is one week. Tasks are taken from the backlog for each sprint iteration. before starting a new sprint there is a discussion meeting to track team progress and test work performance.

Chapter 2

Literature Review

Introduction:

Online learning is the newest and most popular form of distance education today. Within the past decade especially, after the global pandemic of Corona, it has had a major impact on postsecondary education and the trend is only increasing. In our online educational system, students will explore what the experience of online learning is like and how it has changed the role of the instructor. We worked on solving present educational platforms' challenges to achieve effective interaction between the instructor and the student and high education quality through many features that depend on the advantages of artificial intelligence applications with the availability of applications for web and mobile users.

Theoretical Background:

Online learning is education that takes place over the Internet. It is referred to as “e-learning”. However, online learning is just one type of “distance learning”- the umbrella term for any learning that takes place across distance and not in a traditional classroom. Distance learning has a long history and there are several types available today, including

1. **Traditional schooling:** where the students have to attend classes in person.
2. **Correspondence Courses:** conducted through regular mail with little interaction between students and instructors.
3. **Telecourses:** where content is delivered via radio or television broadcast.
4. **CD-ROM Courses:** where the student interacts with static computer content.
5. **Online Learning:** Internet-based courses offered *live* and/or pre-recorded lectures.

By far the most popular approach today is online learning. According to the Sloan Consortium, online enrollments continue to grow at rates faster than for the broader

student population, and institutes of higher education expect the rate of growth to continue increasing.

Some of the key findings:

- Over 1.9 million students were studying online in the fall of 2003.
- Schools expect the number of online students to grow to over 2.6 million by the fall of 2004.
- Schools expect online enrollment growth to accelerate — the expected average growth rate for online students for 2004 is 24.8%, up from 19.8% in 2003.
- The majority of all schools (53.6%) agree that online education is critical to their long-term strategy.
- Since COVID-19 infection, there are currently more than 1.2 billion children in 186 countries affected by school closures due to the pandemic.
- Even before COVID-19, there was already high growth and adoption in education technology, with global tech investments reaching US\$18.66 billion in 2019 and the overall market for online education projected to reach \$350 Billion by 2025.

The previous studies and works

With the improvement in the technology and E-Learning methods have played a vital role changing the atmosphere over the learning paradigms.

During the previous years especially during the covid-19 pandemic, we've found out the importance of online education, It was almost even a substitute for regular education.

According to that, we aim to make a suitable e-learning system that contains:

- A perfect community to reduce the gap between student and teacher.
- An organised content to keep students focused on what they have to do.
- Artificial intelligence technology helps teachers to evaluate students with minimal effort and help students to have an easier and more efficient education journey.

Why online distance learning and why now? Online distance learning meets the needs of an ever-growing population of students who cannot or prefer not to participate in traditional classroom settings. These learners include those unable to attend traditional classes, who live in remote locations, who work full-time and can only study at or after work, and those who simply prefer to learn independently

The minimum requirement for students to participate in an online course is access to a computer, the Internet, and the motivation to succeed in a non-traditional classroom. Online courses provide an excellent method of course delivery unbound by time or location allowing for accessibility to instruction at anytime from anywhere. Learners find the online environment a convenient way to fit education into their busy lives. The ability to access a course from any computer with Internet access, 24 hours a day, seven days a week is a tremendous incentive for many of today's students.

The courses that we have learned from:

1-Durgasoft

Django basic

<https://www.durgasoftonline.com/s/store/courses/description/Django-with-Real-Time-Project-Old-and-New-Videos#tableofcontents>

Django Restapi

<https://www.durgasoftonline.com/s/store/courses/description/Django-REST-API-DRF>

2-Python

<https://www.youtube.com/playlist?list=PL-osiE80TeTt2d9bfVyTiXJA-UTHn6WwU>

Chapter 3

User Experience Design

Materials and Methods

Methods and Procedures:

In order to provide a solution to our end users, we have to follow a process that helps us prioritise the features that solve pain points for them based on evaluative research. Design is based upon an explicit understanding of users, tasks, and environments; is driven and refined by user-centred evaluation; and addresses the whole user experience. The process involves users throughout the design and development process and it is iterative. And finally, the team includes multidisciplinary skills and perspectives.

User-centred design (UCD) is an iterative design process in which designers focus on the users and their needs in each phase of the design process. In UCD, design teams involve users throughout the design process via various research and design techniques, to create highly usable and accessible products for them.

User Centered Design Process



General phases of the user centered design process:

Specify the context of use: Identify the people who will use the product, what they will use it for, and under what conditions they will use it.

In this project, we have two different types of users:

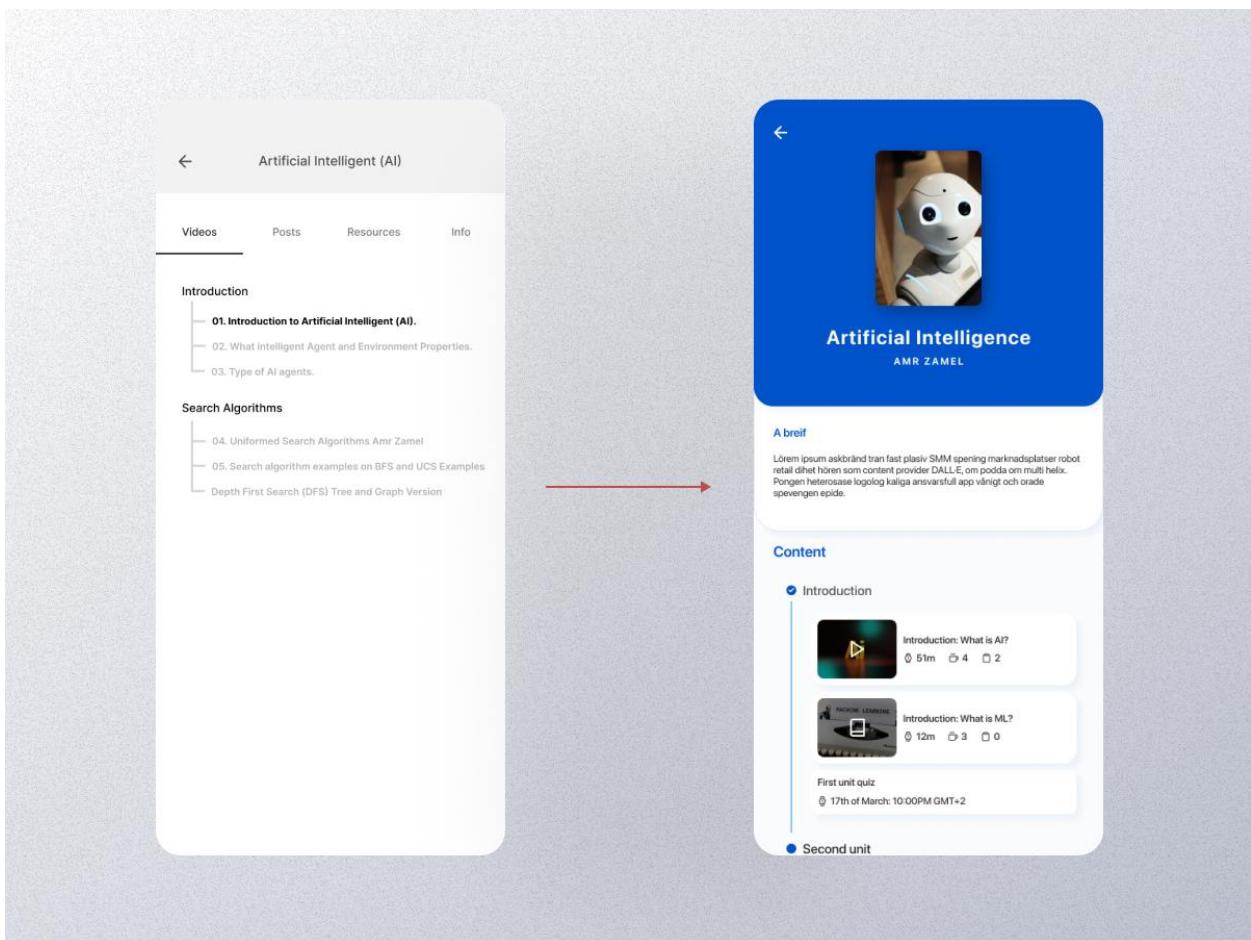
- **Students:** They want to have access to their educational content with a distinctive quality including all related materials directly from the professeurs, be able to ask and get answers to their questions. They want to know the dates of upcoming quizzes/exams and track their performance.
- **Teacher/Professors:** They want a place where they can follow up and present materials to their students, create and schedule exams for them, an easy way to communicate with their them, and answer their questions directly.

Specify requirements: Identify any business requirements or user goals that must be met for the product to be successful.

- Business goal: To make the application widely used across the university, where students can rely on it to do all of their remote activities for all subjects.
- The ultimate user goal: (as mentioned above) is to do his university tasks with ease in one place.

Create design solutions: This part of the process may be done in stages, building from a rough concept to a complete design.

- I've started creating design solutions via rough sketches, then made it lo-fi wireframes.



Demonstrate how the content page evolves from lo-fi to high visual wireframes through an iterative design process.

Header

psum dolor sit amet,
tetur adipiscing elit.
d scelerisque proin vel.

→

Login

Email

Password



Reset your password

Skip →

Login

Don't have an account? Sign up

Header

psum dolor sit amet,
tetur adipiscing elit.
d scelerisque proin vel.

Sign up

Create an account? Login

Register new account

Name

Email

Password



Your Password should be 6 letters or more.

Password

Better use letters, numbers and symbols

Your Password should be 6 letters or more.

Repeat your password

Enter the same previous password

You have read and agree our Privacy Policy

Complete registration

Already have an account? Login

Courses



Recommended

We recommend reviewing the materials below, based on your recent performance in quizzes, assignments...etc.



Artificial Inte

01. Introduction to Artificial Intelligent (AI)



Up next



Artificial Intelligent (AI)

Videos Posts Resources Info

Introduction

— 01. Introduction to Artificial Intelligent (AI).

— 02. What intelligent Agent and Environment Properties.

— 03. Type of AI agents.

Search Algorithms

— 04. Uniformed Search Algorithms Amr Zamel

— 05. Search algorithm examples on BFS and UCS Examples Depth First Search (DFS) Tree and Graph Versions

Recommended

We recommend reviewing the materials below, based on your recent performance in quizzes, assignments...etc.



01. Introduction to Digital

Continue studying



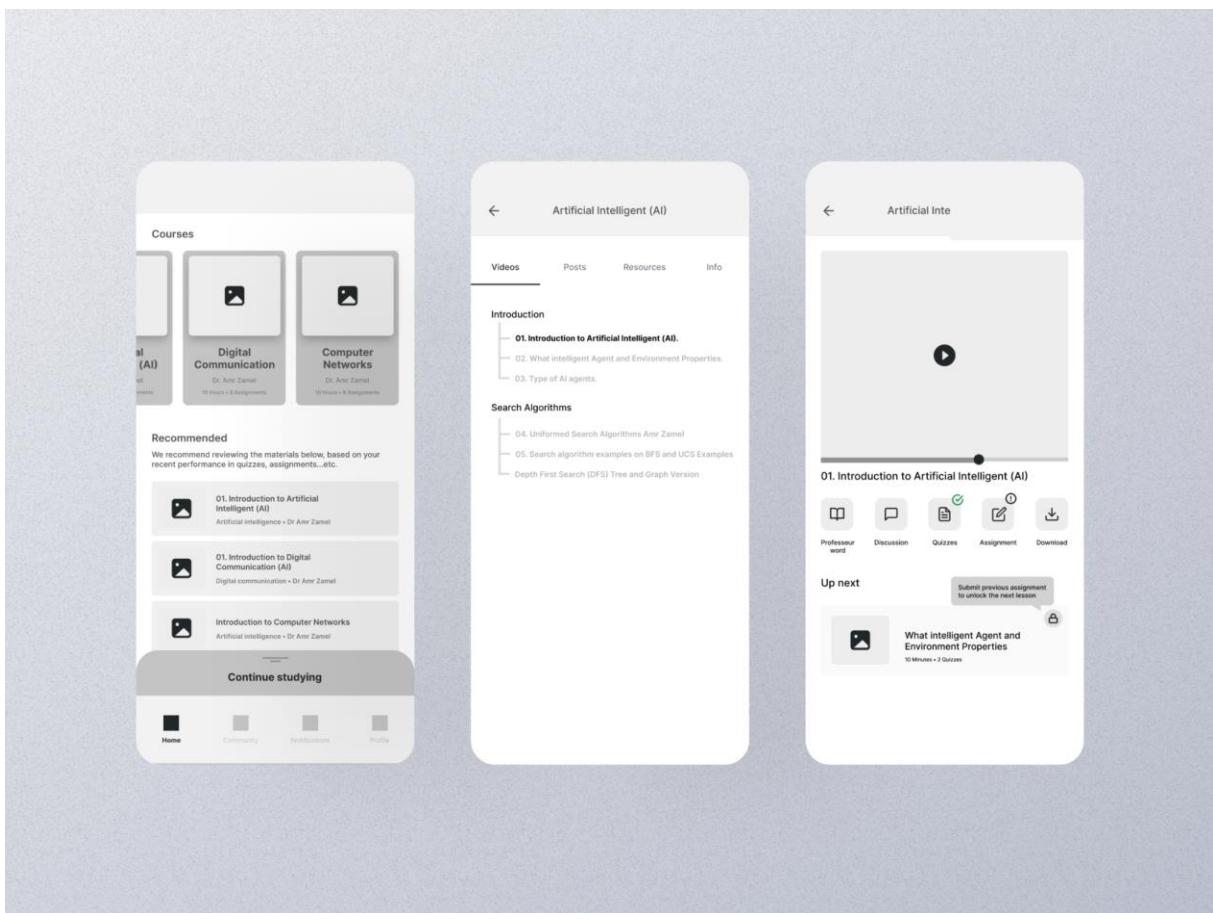
Home Community Notifications

1. What are best models of []?

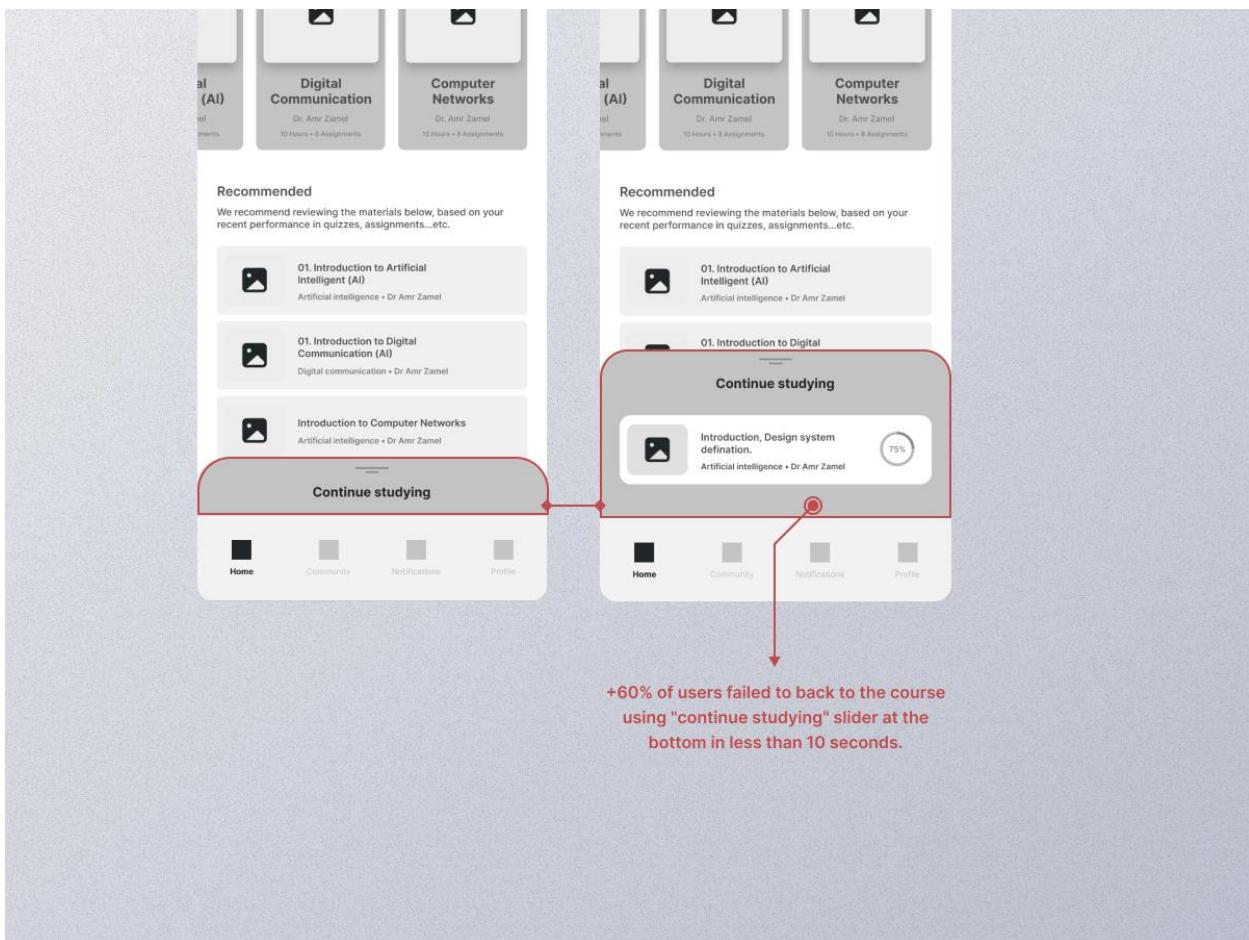
- Option
- Option
- Option
- Option
- Option

Back Next

2. Design strategy

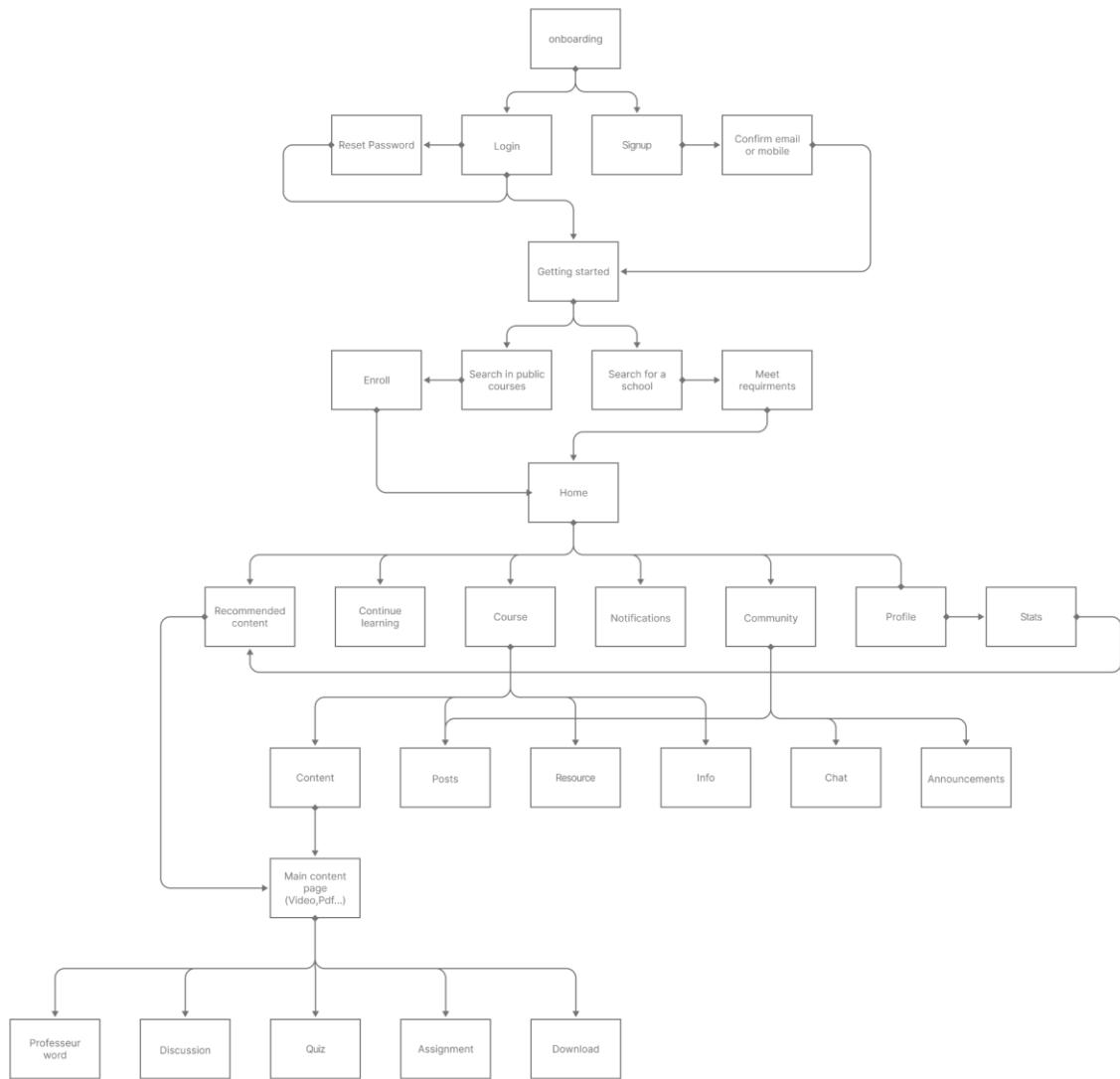


Evaluate designs: Evaluation - ideally through usability testing with actual users - is as integral as quality testing is to good software development.



Materials:

Information Architecture (IA): Information architecture focuses on organizing, structuring, and labeling content in an effective and sustainable way. The goal is to help users find information and complete tasks.

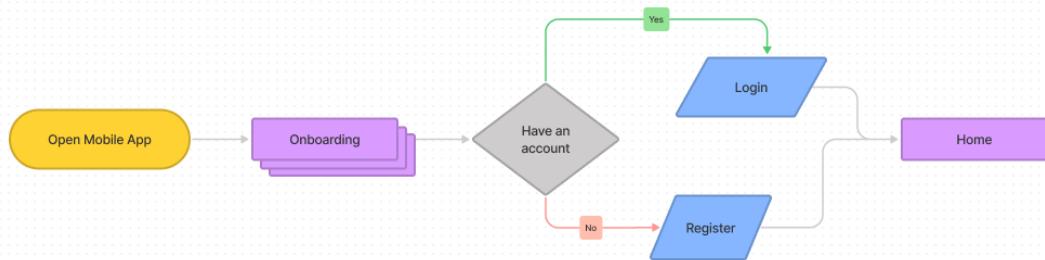


AI

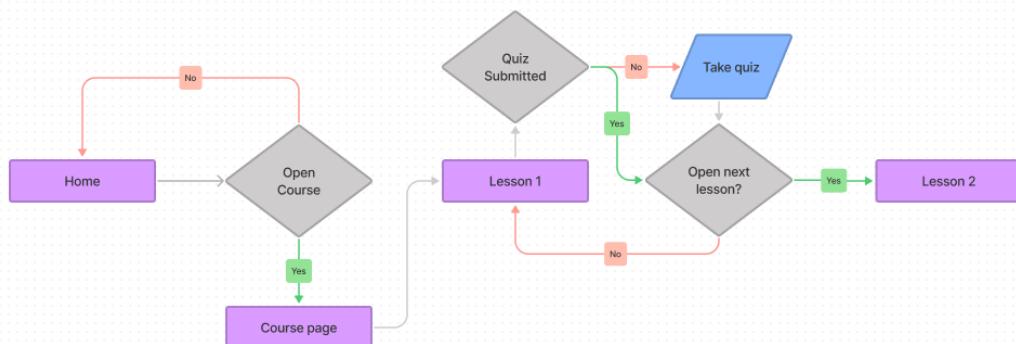
for mobile app and website.

User flows: User flow is the path taken by a prototypical user on a website or app to complete a task. The user flow takes them from their entry point through a set of steps towards a successful outcome and final action, such as purchasing a product.

Flow 1: Onboarding new users

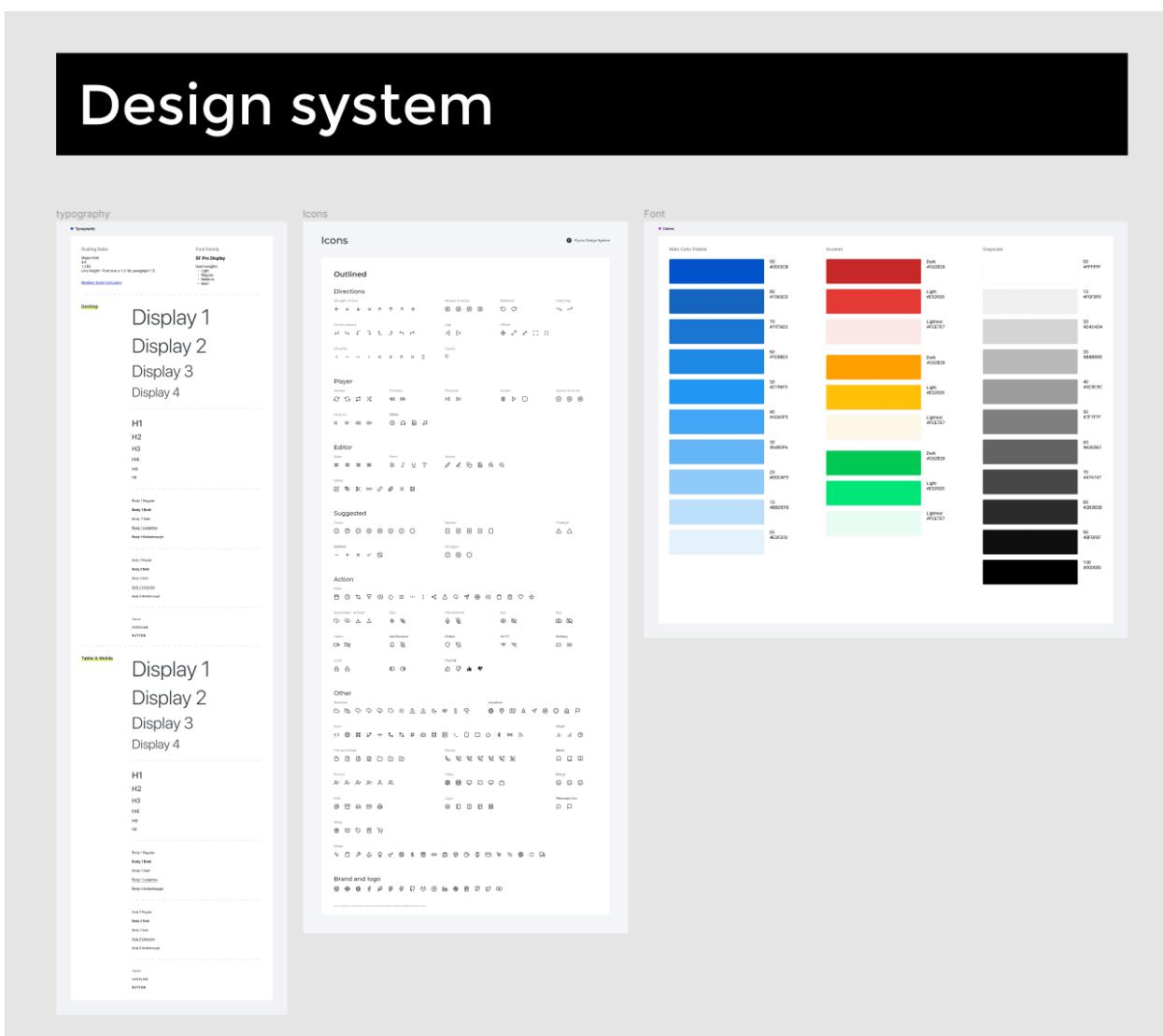


Flow 2: View course content lesson by lesson



Design System: A design system is a complete set of standards intended to manage design at scale using reusable components and patterns.

We created a design system to help create visual consistency across our product and help hand off to developers.



A set of typography, core colours with their variants, and icons.

- Programs:**

Pen and paper: For Ideation, and build rough sketches.

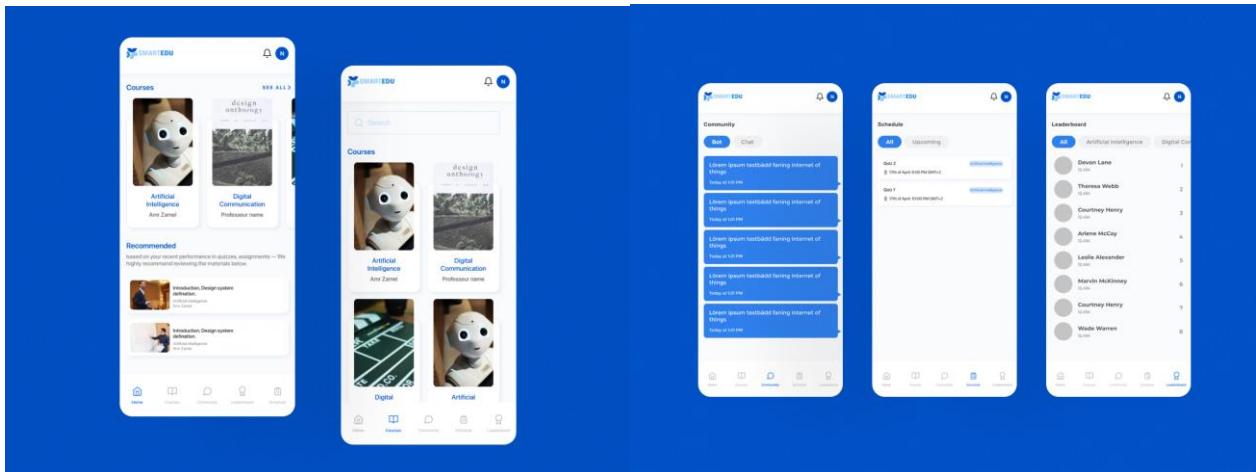
Figma: For overall low and high visual design, make design systems, and handoff designs to developers.

Results and Discussion

Onboarding: To educate the student about the objectives of the app, and to provide an easy way for them to authenticate.



Nav pages:



Chapter 4

Website Application

4.1 FrontEnd

4.1.1 Framework & programming language:

In this project we use HTML , CSS ,JavaScript, JQuery, and Bootstrap.

1- HTML 5:

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages.
- HTML describes the structure of a Web page.
- HTML consists of a series of elements.
- HTML elements tell the browser how to display the content.
- An HTML element is defined by a start tag, some content, and an end tag:
`<tagname> Content goes here... </tagname>`
- The HTML element is everything from the start tag to the end tag: `<h1>My First Heading</h1>`

2- CSS 3:

- CSS stands for Cascading Style Sheets
- CSS is the language we use to style an HTML document.
- CSS describes how HTML elements are to be displayed on screen.
- CSS saves a lot of work. It can control the layout of multiple web pages all at once
- External stylesheets are stored in CSS files
- CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

3- JavaScript :

- JavaScript is the world's most popular programming language.
- JavaScript is the programming language of the Web.
- Javascript Can Change HTML Content
- JavaScript Can Change HTML Attribute Values
- JavaScript Can Change HTML Styles (CSS)
- JavaScript Can Hide HTML Elements
- JavaScript Can Show HTML Elements

4- jQuery:

- jQuery is a JavaScript library.
- jQuery is a lightweight, "write less, do more"
- jQuery greatly simplifies JavaScript programming
- The purpose of jQuery is to make it much easier to use JavaScript on website pages.
- jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.

5- BootStrap 5:

- Bootstrap is a free front-end framework for faster and easier web development
- Bootstrap includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels, and many others, as well as optional JavaScript plugins
- Bootstrap also gives you the ability to easily create responsive designs
- Advantages of Bootstrap:
 - **Easy to use:** Anybody with just a basic knowledge of HTML and CSS can start using Bootstrap
 - **Responsive features:** Bootstrap's responsive CSS adjusts to phones, tablets, and desktops

- **Browser compatibility:** Bootstrap is compatible with all modern browsers (Chrome, Firefox, Internet Explorer, Edge, Safari, and Opera)
- **We used Bootstrap 5.** Bootstrap 5 is the newest version of Bootstrap, with new components, faster stylesheets, more responsiveness, etc. It supports the latest, stable releases of all major browsers and platforms.

4.1.2. User Flow Diagram:

The user can be: Student or teacher.

Our goals:

For the teacher:

Its work is facilitated through:

- Easy working test.
- Easy to perform assignments, whether programming or otherwise
- Reduce fraud as much as possible and help detect fraud in cases of student similarity.
- Upload each course in the order of lectures and each lecture has its own.

For the students:

- Easy access to all assignments and competitions through the schedule page.
- Student assistant in obtaining information through the chat room and chat bot.
- Stimulate it through the leaderboard.
- It is easy to send a question or comment about a unit in this course.
- Easy students communicate with each other in the chat room with each other and their assistants.

To achieve this goal, we used the diagram in figure 4.1.

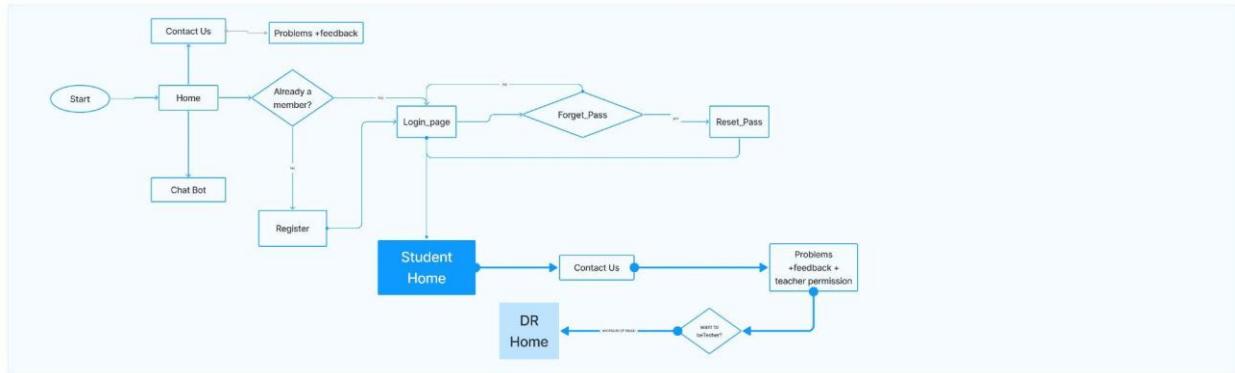


Figure 4.1 User Flow Diagram

Explanation of Figure 4.1:

In Figure 4.1, we find that there are 3 sections:

1. Public home with Authentication
2. Home for Students
3. Home for Teacher (DR)

As shown in figure 4.1, after logging in (all users register as students).if the user need to teach on our website he can contact us.

First: Public home with Authentication

Anyone visiting our website can reach the public home, which contains a navbar containing a login button and a start to register button. As shown in figure 4.2.

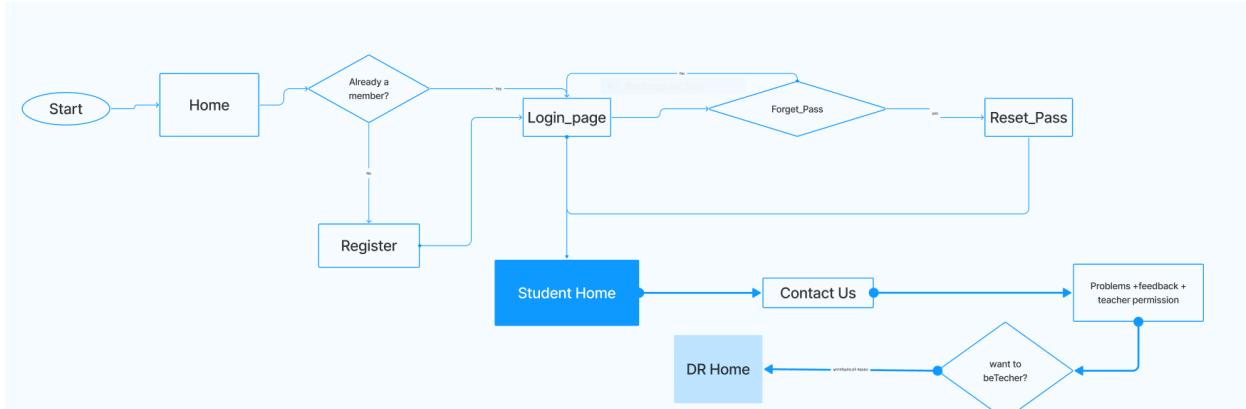


Figure 4.2 Authentication

Second: Home for Students

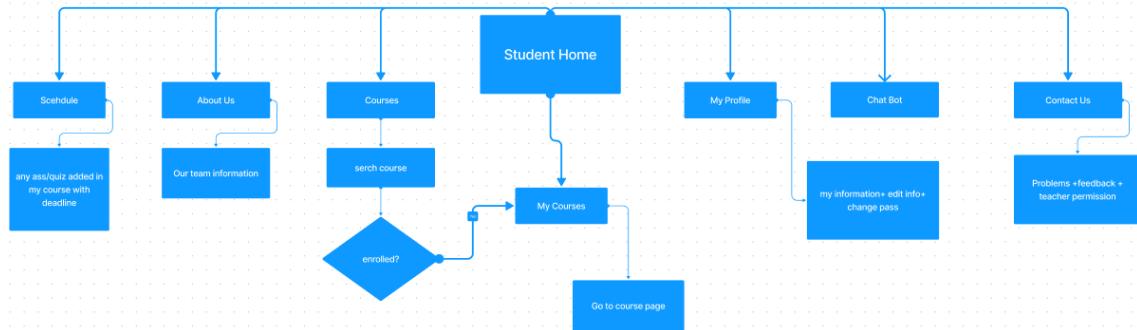


figure 4.3 student flow diagram

As shown in figure 4.3, when the student in his home he can see:

- ◆ the student's **home**
- ◆ The **chatbot** where He can get information from.
- ◆ He can **contact us** if he has a problem, feedback and if he is a teacher.
- ◆ **My profile** page contain his information ,edit his profile and change password
- ◆ **About us** which contains our team information.
- ◆ **Courses**: He can also search for courses and enroll in any course if he has a course password.
- ◆ **My courses** page contains any registered course.
- ◆ **Schedule** page contains assignments and quizzes for courses with a deadline.

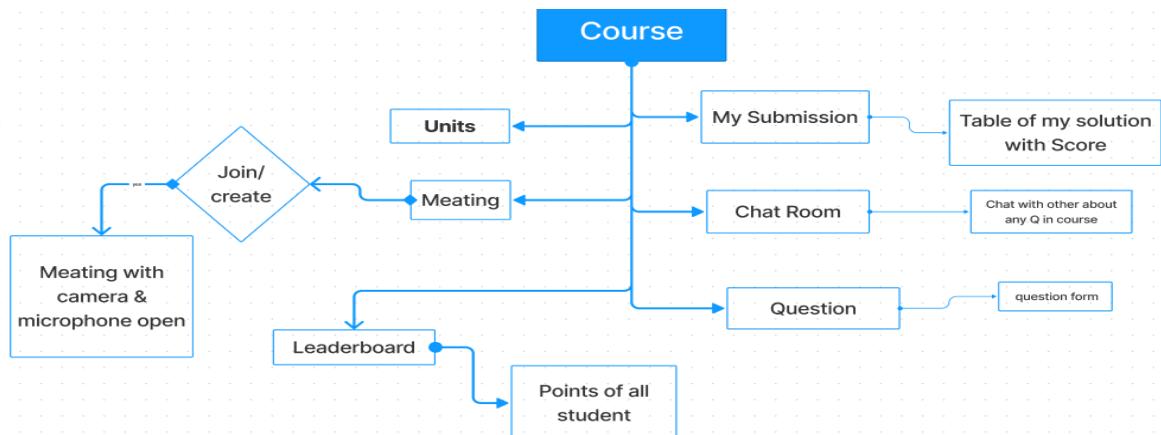


figure 4.4 flow of student course

As shown in figure 4.4. From my course page, the student can access any course in which to register, which contains:

- **Units** (block we will focus it after this)
- **My submission** page that contains his solution to the assignment with grades
- **leaderboard** page that contain all points of students
- **chat room page** which he can communicate with other
- **Meeting:** he can join or create meeting
- **Question page:** ask any questions about the course.

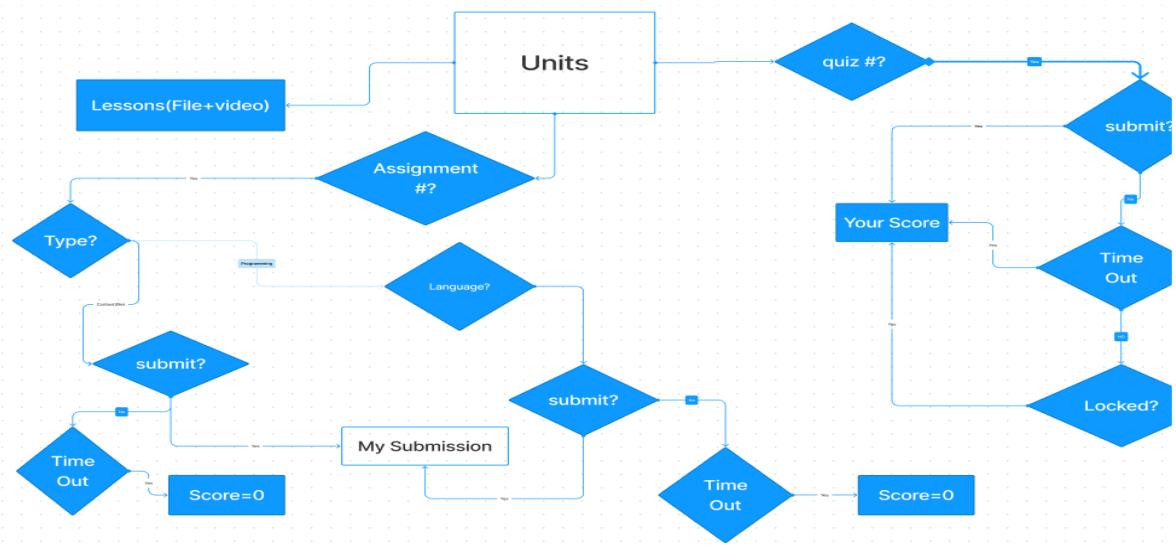


figure 4.5 flow of units in course page

As shown in figure 4.5. the student can access any unit in his course, which contains:

- **Lesson / Lecture file and video.**
- **The assignment page contains** the assignment to be delivered for the student to download and send its solution through the form with the possibility of sending a comment. If the assignment time expires, the student appears expired and finds a button to start the assignment disabled.
- **The quiz page** contains the name with the start time and the dead time but can not be opened until the start of the exam because the opening button is disabled. But when the test begins, he finds that the start button is open.

During the test, the student finds a form with questions with a counter at the top of the page to know the rest of the time is over. If he tries twice to get out of the page, he shows him a warning to stop cheating so that he does not cancel the exam, and in the third attempt we tell him that he was cancelled, cancels the quiz and sends his solution. This is known as **locked the quiz** and I implemented it by js and we will explain the sample code of this after the flow diagram.

Third: Home for Teacher

After the admin accepted the request user to be a teacher, we opened the teacher's home for the user.

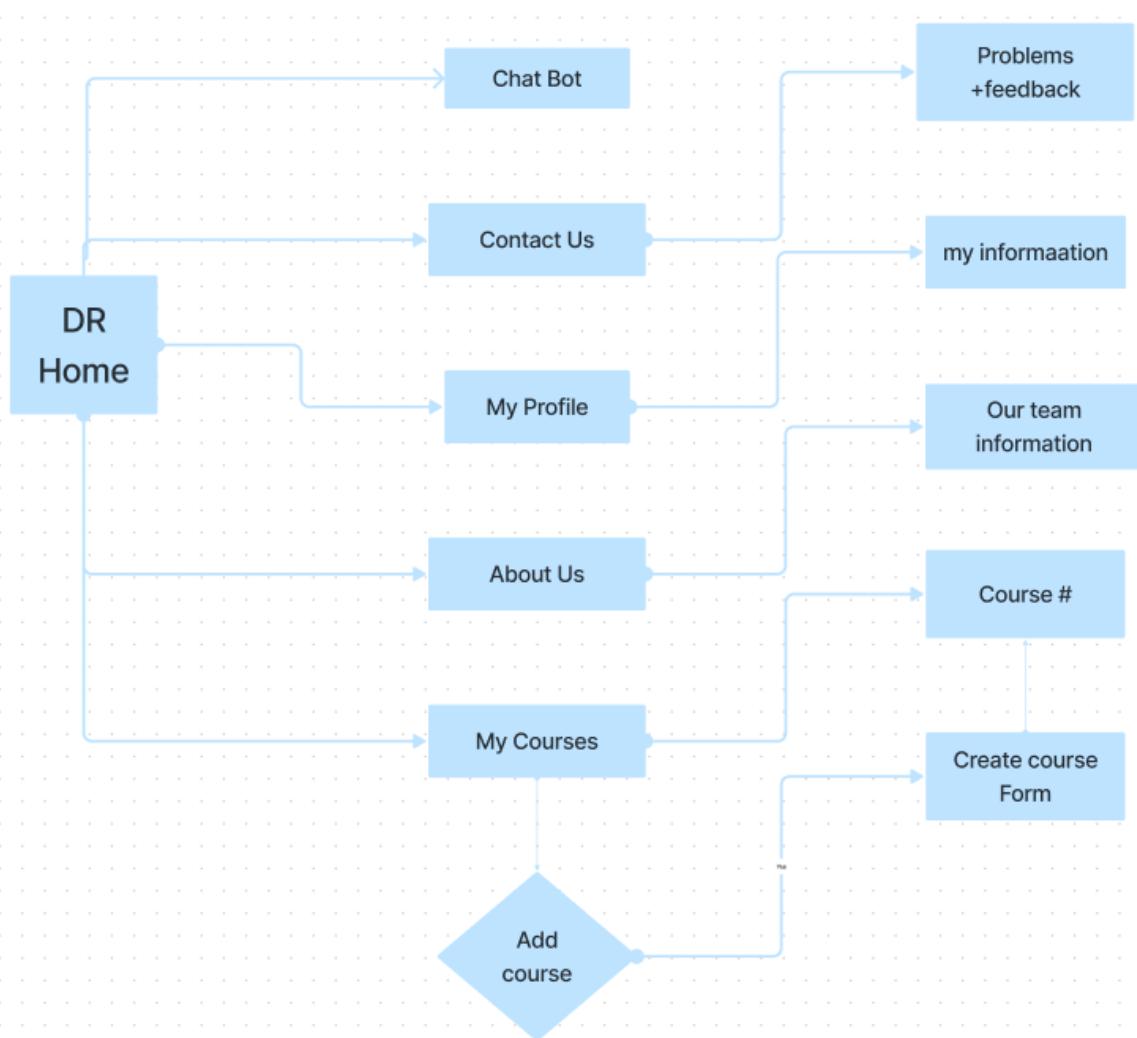


Figure 4.6 the teacher's flow

As shown in figure 4.6, In the teacher's home , he can see:

- ◆ The **chatbot** where He can get information from.
- ◆ **About us** which contains our team information.
- ◆ He can **contact us** if he has a problem or feedback.
- ◆ **My profile** page contains his information ,edit his profile and change his password.
- ◆ **My courses** page contains the courses that were created by him and he can add more from the add button and Fills the form.

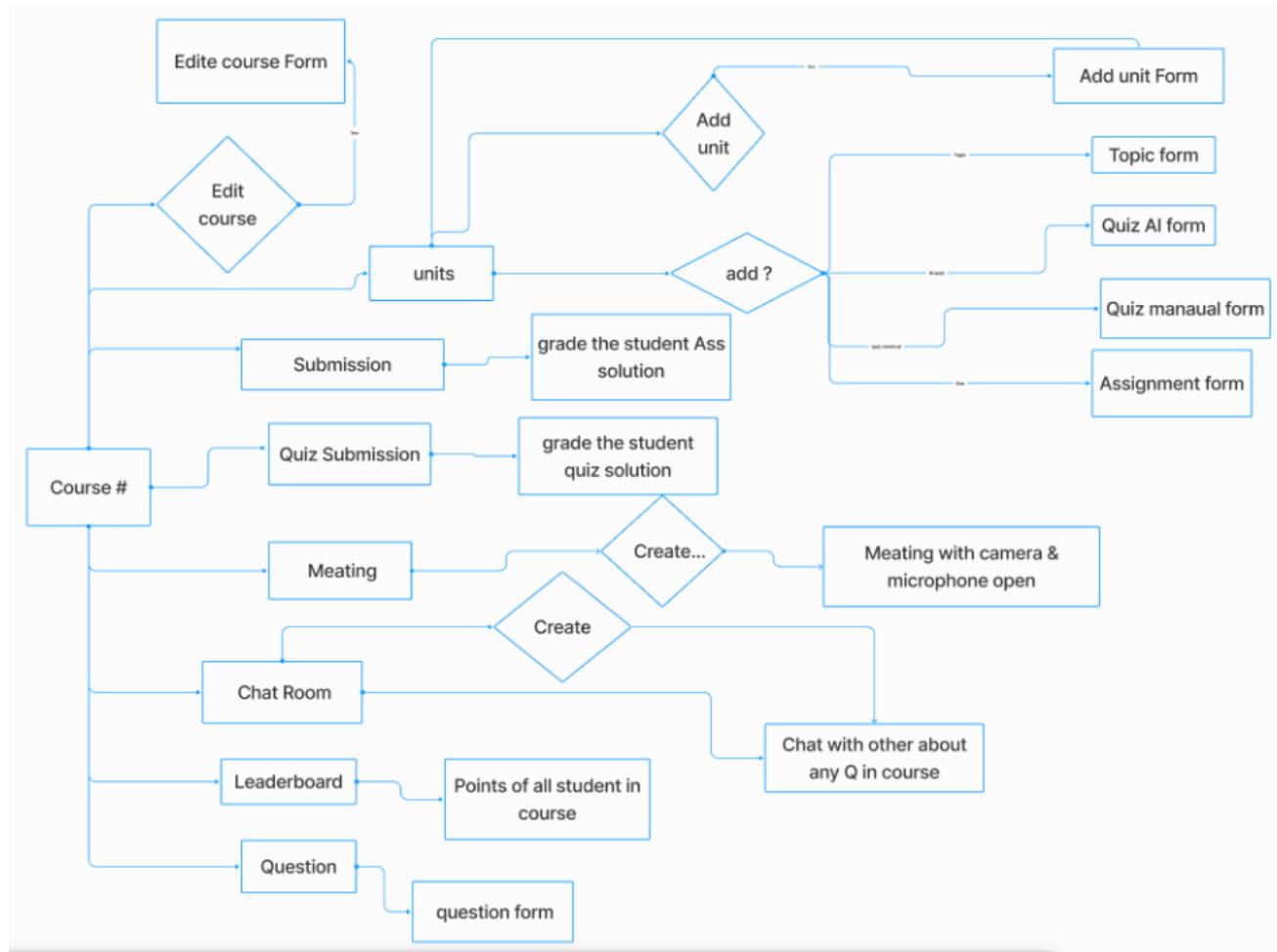


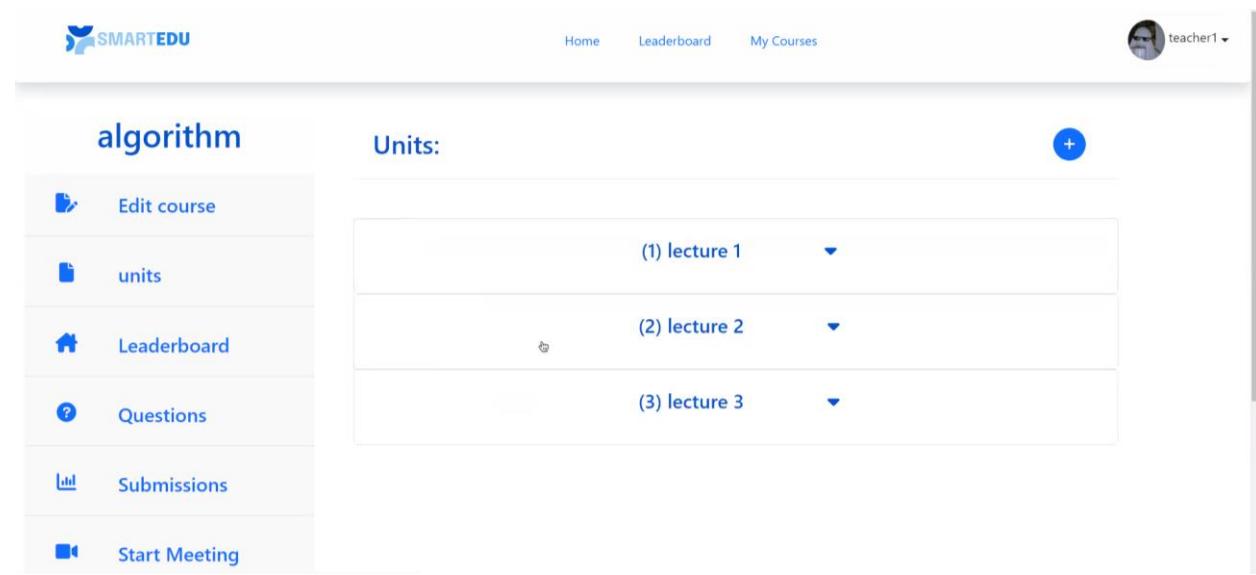
Figure 4.7 the teacher's course flow

As shown in figure 4.7, In each course page, he can:

- **Edit** his course or **delete** it.
- See his **units**, each unit has:
 - **Topic form** to add lecture video and files.
 - **Add Assignments**: programming or regular.

- **Add quizzes: manual or by AI.**
- **Add a unit** by filling its form if needed.
- The **submission** page that contains the student's solution to the assignments is graded.
- The **Quiz submission** page contains the student's solution to the quizzes graded.
- **leaderboard** page that contain all points of students
- **chat room** page that he can use to communicate with others.
- he can join or create **meeting**
- ask any **questions** of course.

We implemented the page of doctor too simple to find all his needed in it for example



The screenshot shows the SMARTEDU teacher dashboard. At the top, there is a navigation bar with links for 'Home', 'Leaderboard', 'My Courses', and a user profile for 'teacher1'. On the left, a sidebar menu for the course 'algorithm' includes options: 'Edit course', 'units' (which is currently selected), 'Leaderboard', 'Questions', 'Submissions', and 'Start Meeting'. The main content area is titled 'Units:' and displays three units: '(1) lecture 1', '(2) lecture 2', and '(3) lecture 3'. A blue '+' button is located at the top right of the units list.

Figure 4.8 the Course units

As shown in Figure 4.8, we executed a very simple teacher page to find everything he needed in it. If the teacher clicks on the chat room, the units hide(the page in right) and show the chat room in place as shown in figure 4.9, and so on.

Figure 4.9 the teacher's room in course

I locked the quiz :

If the student tries to open any page during the quiz, we show him an alert to stop cheating and give him only two chances as shown in figure 4.10, and in the third try the exam closes and sends what the student did before the lock and the page of quiz with his score will appear as shown in figure 4.11
During the test, we display the remaining time to the deadline at the top of the test page so that the student can know the remaining time and send before it expires. As shown in figure 4.10.



Figure 4.10 quiz form page

The screenshot shows a quiz interface with the following details:

QUIZ		SUBMITTED
Quiz Name	quiz	
Submit Start Time	July 13, 2022, 11:33 a.m.	
Submit End Time	July 13, 2022, 12:33 p.m.	
student1 score	Scored: 2	

[Cancel](#)

Figure 4.11 quiz page locked

And it's implemented by js.

To calculate the time we get the time from the teacher and through the back end we get the deadline and start time of the quiz and we calculate the difference between the current time and the initial time of the test so the test begins when they are equal and when the test opens show the difference between the current and final time at the top of the page and the when of the arrival of the current time of the final closes the test and we find the start button disabled. As shown in figure 4.12.

The screenshot shows a quiz interface with the following details:

QUIZ		NOT SUBMITTED
Quiz Name	quiz1	
Submit Start Time	July 8, 2022, 6:23 p.m.	
Submit End Time	July 8, 2022, 6:24 p.m.	
student1 score		

[Start Quiz](#) [Cancel](#)

Figure 4.12 quiz page after deadline

The same is done in the case of assignments as shown in figure 4.13

ASSIGNMENT

NOT SUBMITTED

Assignment Name	assignment1
Submit Dead Time	July 13, 2022, 1:09 p.m.
Assignment Type	Regular_File
Language Type	None
Points	5
File in this assignment	fromzagazig.pdf 
Points	see your grades

[Start Assignment](#) [Cancel](#)

Figure 4.13 Expired assignment

4.2-Backend

4.2.1)Framework & programming language:

The first step was to choose a good, reliable language and a framework for the implementation of our website, we selected Django framework. Django Web framework have built in high-level Python language that allows fast development.

4.2.1.1)Why did we use Django web Framework?

- 1-Django is a free and open-source web framework.
- 2-It is used by several top websites like Youtube, Google, Dropbox, Yahoo Maps, Mozilla, Instagram, Washington Times, Nasa and many more
- 3-Django was designed to help developers take applications from concept to completion as quickly as possible.
- 4-Django takes security seriously and helps developers avoid many common security mistakes, such as SQL injection, cross-site scripting, cross-site request forgery and clickjacking. Its user authentication system provides a secure way to manage user accounts and passwords, avoiding common mistakes like putting session information in cookies where it is vulnerable (instead cookies just contain a key, and the actual data is stored in the database) or directly storing passwords rather than a password hash.

4.2.1.2)Which design pattern does Django Follow? or What does django code look like? or What is Dango Framework process?

Django follow the **Model View Template** design pattern!!

In a traditional data-driven website, a web application waits for HTTP requests from the web browser. When a request is received the application works out what is needed based on the URL and possibly information in POST data or GET data. Depending on what is required it may then read or write information from a database or perform other tasks required to satisfy the request. The application will then return a response to the web browser which will be in HTML form. which is nothing but the user interface.

Django web applications typically group the code that handles each of these steps into separate files: model.py file & view.py file & template html files & url.py file!!

1-model.py: As the part of web application development, compulsory we required to interact with database to store our data and to retrieve our stored data,Django provides a big in-built support for database operations. Django provides one inbuilt database sqlite3,For small to medium applications this database is more than enough. Django can provide support for other databases also like oracle, mysql,postgresql etc In this file we have to store the application's database, and in django we only write the database as a python class, and django will convert it into sql code, each python class maps to one database table.

2-view.py: In this file we have to save functions that handles requests and return required responses, these function can take data from models(which is nothing but the database) then do business logic on it then send it to the template(which is nothing but html)

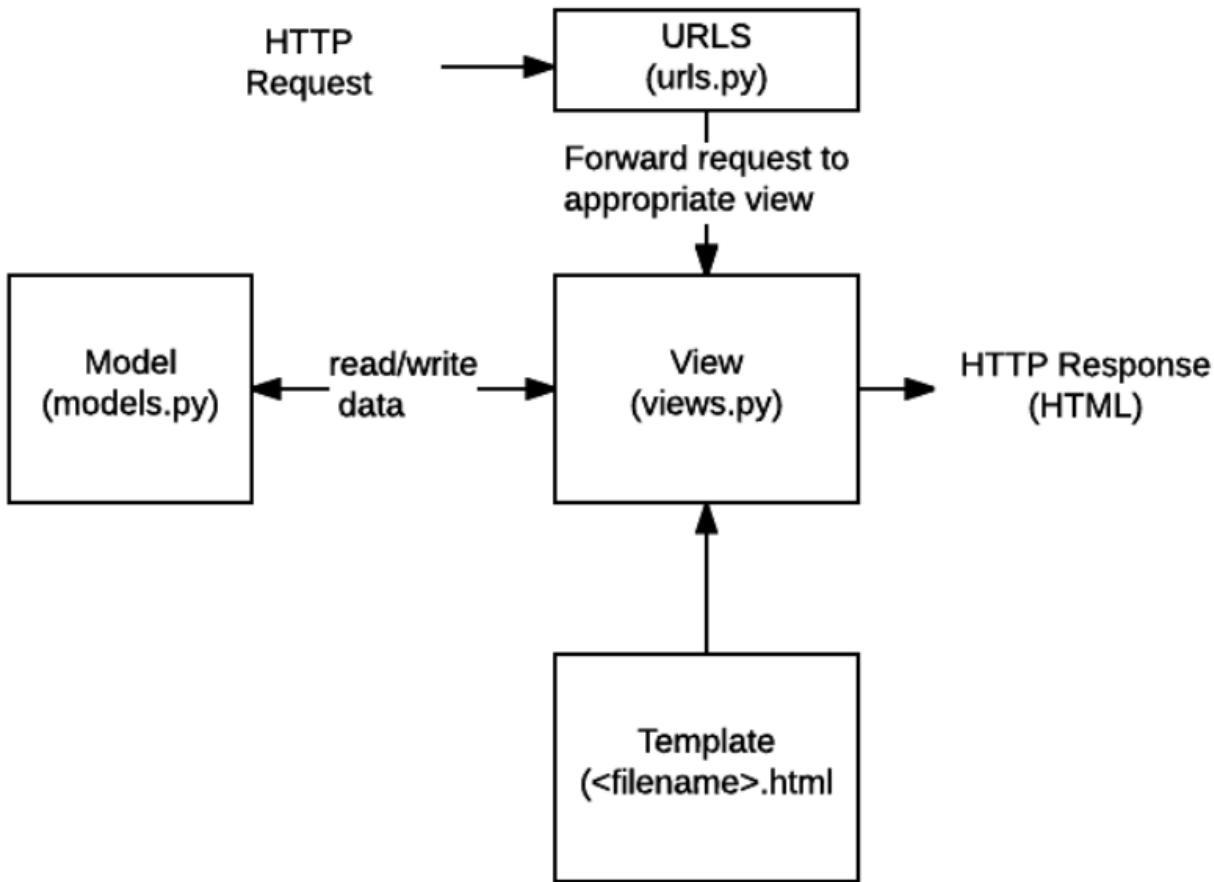
3-urls.py: In this file we have to save all url patterns, These url-patterns will be used by end-user to send request for our views, and the url will make connection between the template(html) and the view(functions), each url pattern will refer to one view function.

In short: (MVT)

1-Model: is the database, we only write the database as python class and it will be converted into sql code programmably.

2-View : Is the business logic or the operations that we do in the database, Each view will be specified as one function in views.py file, Each view should return HttpResponse object with our required response.

3-Template: is the html files that show the data from the database via the view function by using the url pattern.



1. Whenever the end user sends the request ,the first Django development server will get that request.
2. From the Request django will identify the url pattern and by using urls.py, the corresponding view will be identified.
3. The request will be forwarded to the view. The corresponding function will be executed and provide the required response to the end user in the form of HTML.

Role of Web Server:

- Web Server** provides an environment to run our web applications.
- Web Server** is responsible to receive the request and forward request to the corresponding web component based on url-pattern and to provide response to the end user.
- Django framework** is responsible for providing development servers. Even Django framework provides one inbuilt database sqlite. Special Thanks to Django.

4.2.2)Database

introduction

What is data? Data is nothing but the information that you can store. It can be used in a variety of forms like text, numbers, media, etc. it can be stored in pieces of paper or electronic memory, etc.

What is a database?

A database is an organised collection of data, so that it can be easily accessed and managed.

You can organise data into tables, rows, columns.

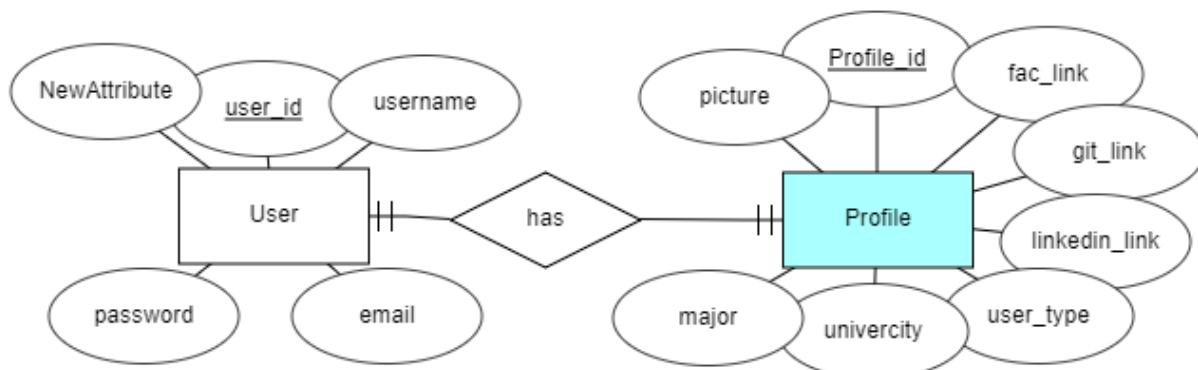
The main purpose of the database is to operate a large amount of information by storing, retrieving, and managing data.

overview of our Database & The ERD diagram:

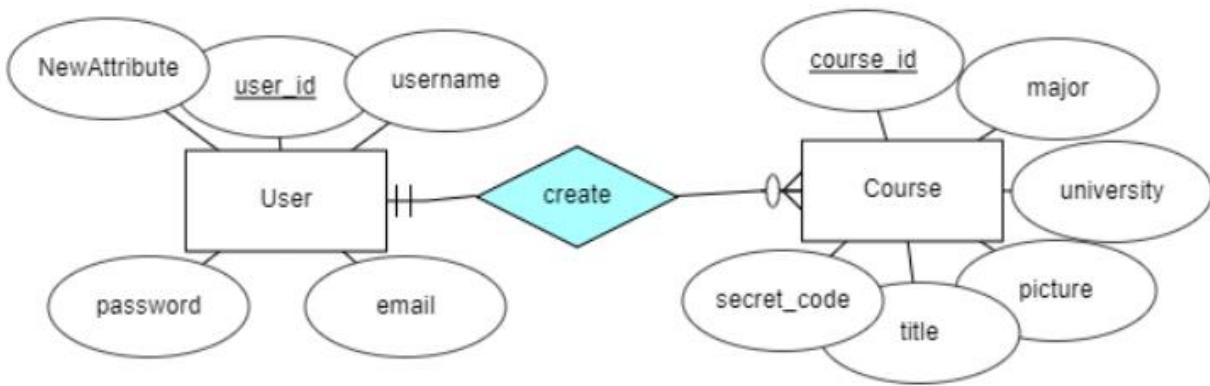
First of all, we will analysis the project by writing the business rules, then we will define all database Tables(in django table refer to python class) and spread all Entities(in django an entity refer to python class name) and all the attributes of these entities(in django attribute refer to python variable)

The Business Roles:

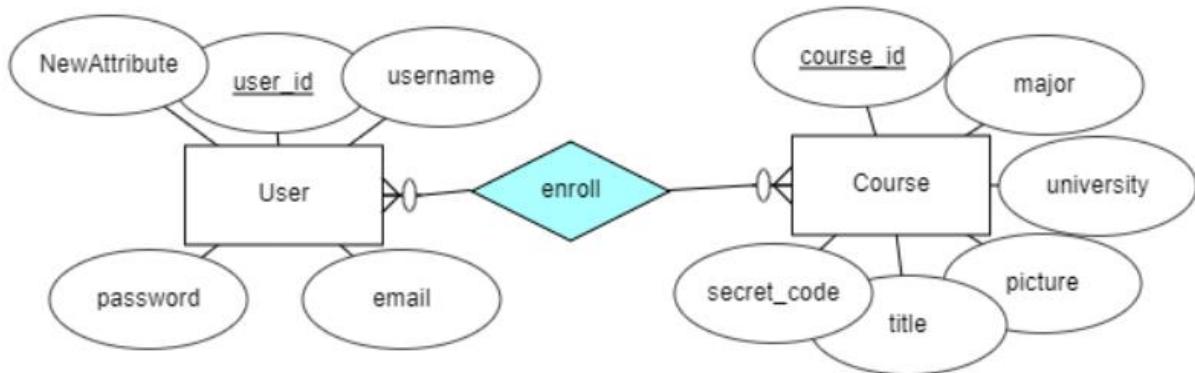
1-Each user has only one profile, in profile we will decide the user type(teacher/student), the default user type will be (student type) and the admin can decide if the user can be teacher or not.



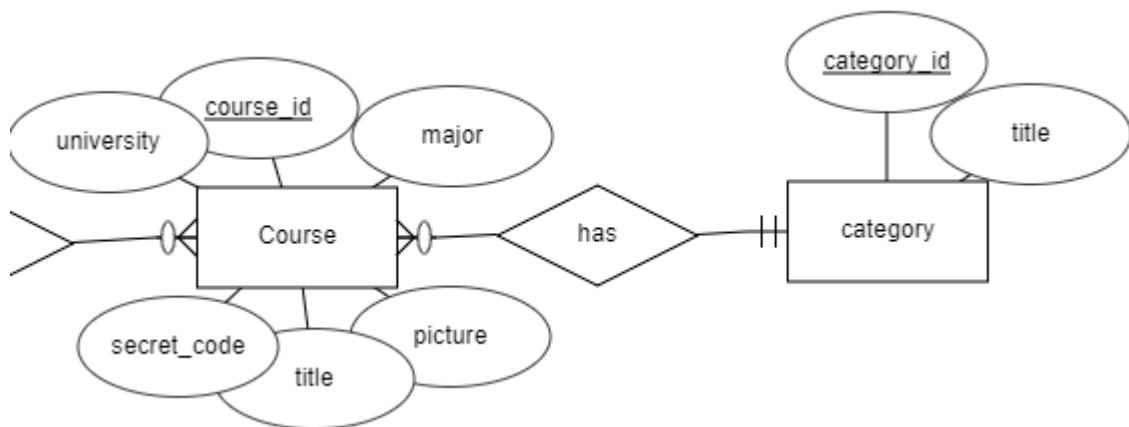
2-each user(teacher type) can create many courses, and each course must created by only one user(teacher type)



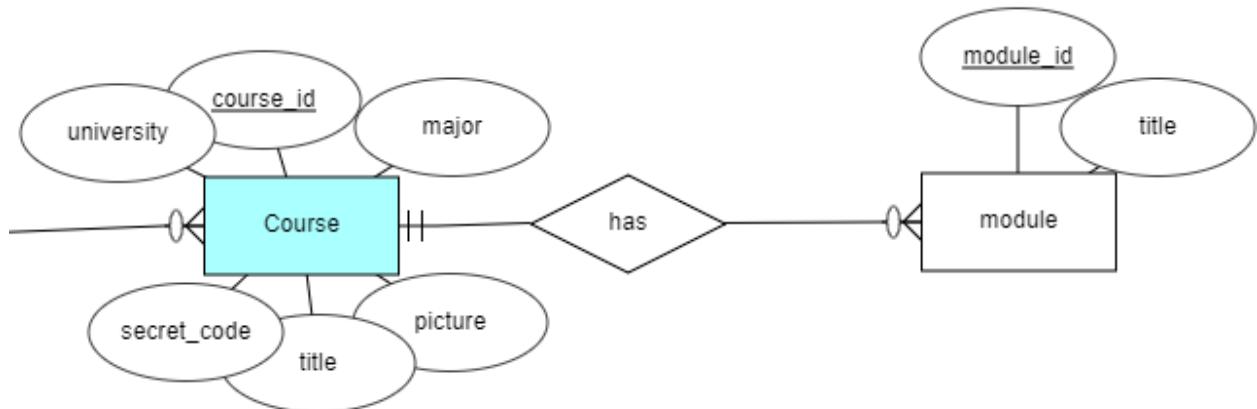
3-many users(student type) can enrol many courses,each course can contain enrolled user(student type) or not



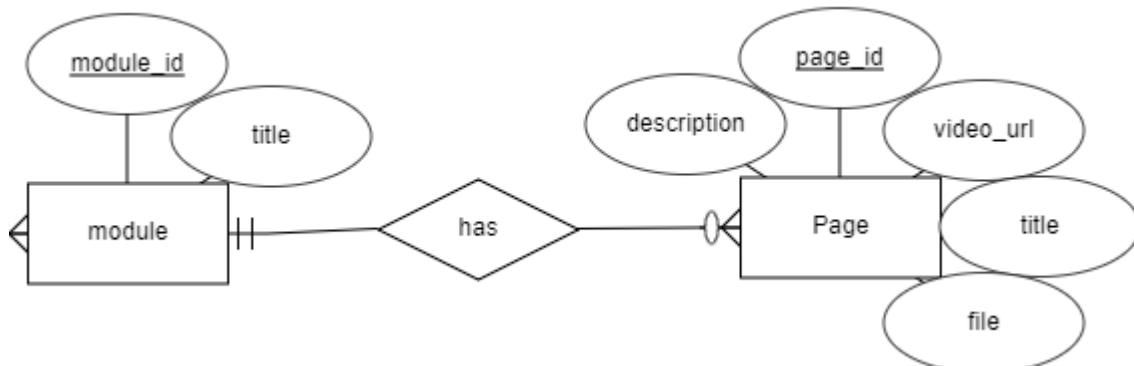
4-each category may has many courses,each course must has only one category



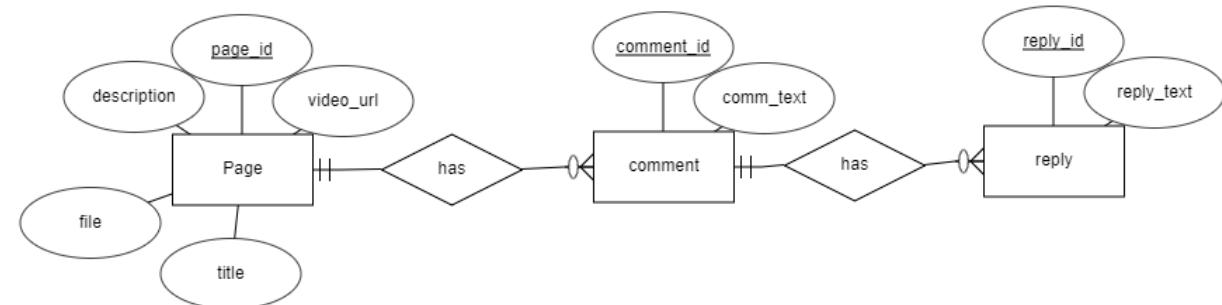
5-many courses may have one or more modules(it also known as lecture or topic or chapter)



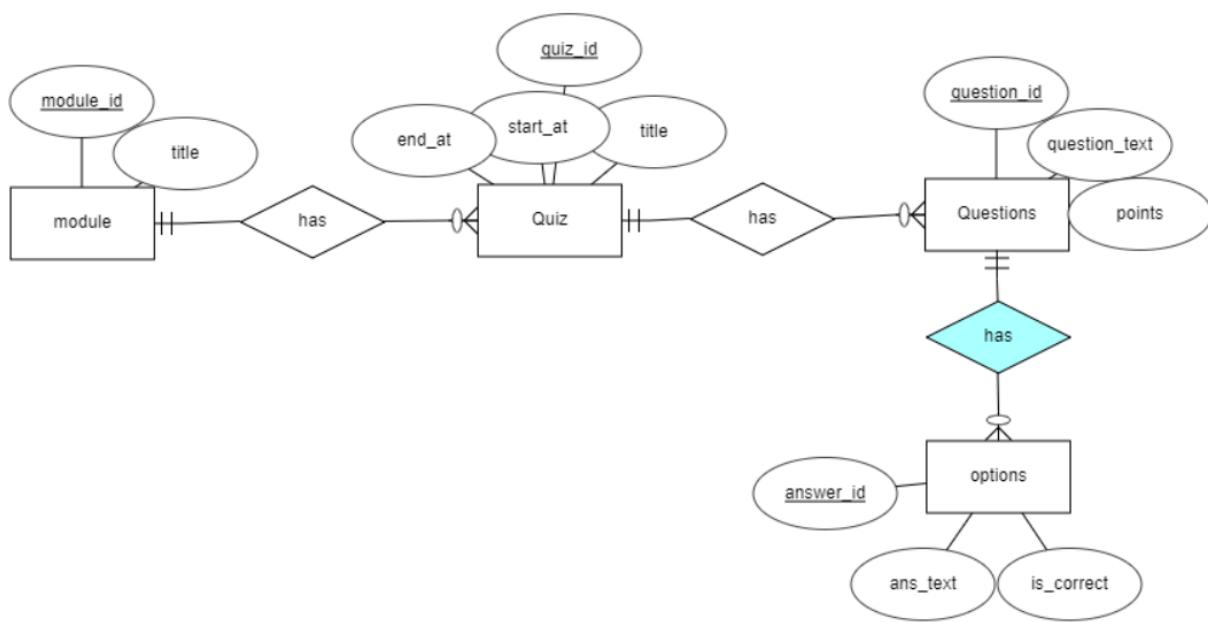
6-each module can contain one or more page(video) &one or more assignment&one or more quiz



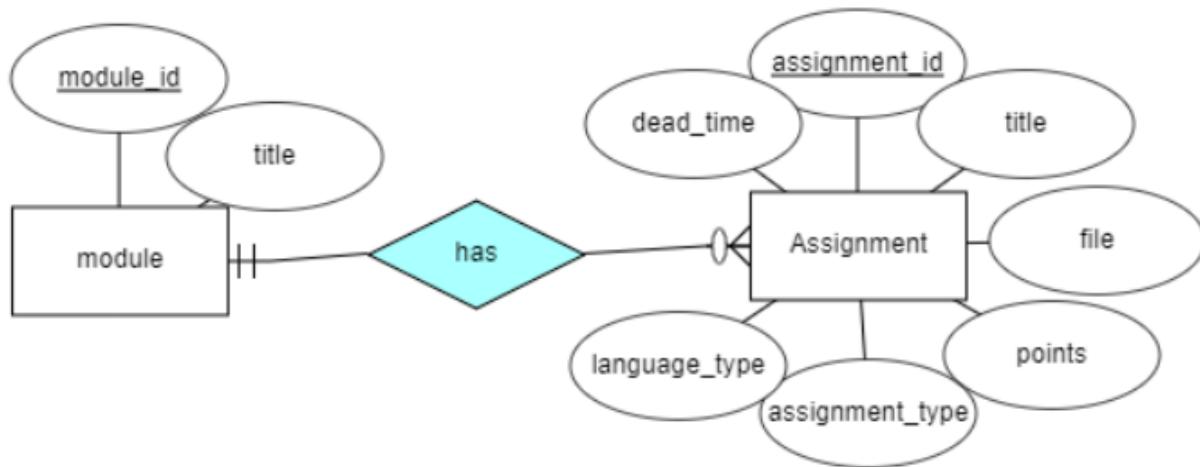
7-each page can contain one or more comments and each comment can contain one or more replies and each user(teacher or enrolled student) can write one or more either comment or reply



8-each module can contain one or more quiz,each quiz can be created by one user (teacher type)and can be attimbed by many users(student type),each quiz can contain many questions,each question can contain many answers.



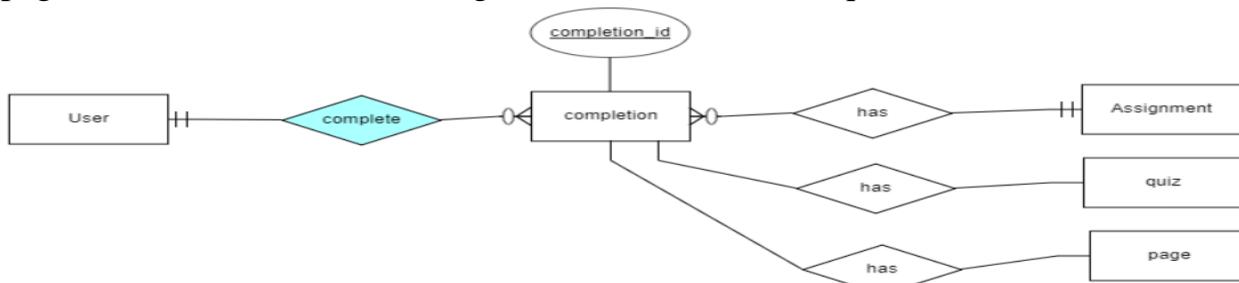
9-each module can contain one or more assignments,each user(teacher type) may create one or more assignments,each assignment must have one user(teacher type).



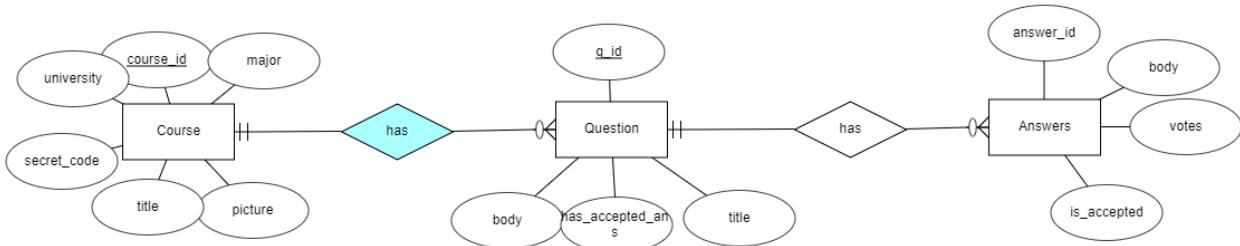
10-each user(student type) may submit one or more assignments,each assignment can be submitted by at least one user(student type)

11-each user(teacher type) may grade one or more submitted assignments

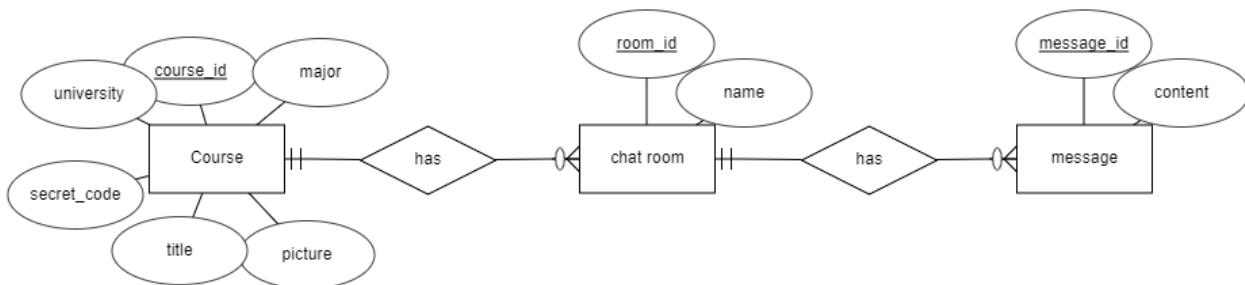
12-each user(student type) can complete(which means mark as done) one or more page(video) & one or more assignments & one or more quizzes



13-each course can have one or more question(ask section almost like stackoverflow) and each question may have one or more answers,each user(teacher or enrolled student) can ask one or more questions & each user(teacher or enrolled student) can answer one or more answers.



14-each course can have one or more chat room ,each user(teacher type) can create one or more chatroom in each course,each user(student type) can join one or more chatroom in the course that he/she enrol,each chat room can have one or more messages and each joined user can write one or more messages.



15-if the user(teacher type) create a quiz or an assignment,the enrolled users(students) will get a notification in schedule section.

The Entities & attributes:

- 1-**User**(id(pk),username,email,password1,password2)
- 2-**Profile**(id(pk),**user(fk)**,picture,phone_number,college_id,university_name, Major_type,user_type[instructor-student],points,facebook_link,github_link, linkenin_link)
- 3-**ContactUS**(id(pk),name,email,message)
- 4-**Category**(id(pk),title)
- 5-**Course**(id(pk),picture,title,secret_code,description,university_name,major_type, category(fk),user_teacher(fk),user_students(fk),**modules(fk)**,questions(fk))
- 6-**Grade**(id(pk),course(fk),submission(fk),points,graded_by(fk))
- 7-**Assignment**(id(pk),title,points,assignment_type,language_type,dead_time, file,user(fk))

8-Submission(id(pk),file,assignment_type,language_type,comment,date, assignment(fk),user(fk))

9-RoomMember(id(pk),roomname,name)

10-Completion(id(pk),user(fk),course(fk),page(fk),quiz(fk),assignment(fk))

11-Module(id(pk),title,user(fk),pages(fk),quizzes(fk),assignments(fk))

12-Page(id(pk),title,video_url,description,file,user(fk))

13-Comment(id(pk),user(fk),page(fk),comment_text)

14-Reply(id(pk),user(fk),page(fk),comment(fk),reply_text)

15-Question(id(pk),user(fk),title,body,has_accepted_answer)

16-Answer(id(pk),user(fk),question(fk),body,votes,is_accepted_answer)

17-Votes(id(pk),user(fk),answer(fk),vote)

18-Quizzes(id(pk),user(fk),title,start_at,end_at,questions(fk))

19-Questions(id(pk),user(fk),answers(fk),text,points)

20-Answers(id(pk),user(fk),answer_text,is_correct)

21-Attemper(id(pk),user(fk),quiz(fk),score)

22-Attempt(id(pk),attemper(fk),quiz(fk),question(fk),answer(fk))

23-Room(id(pk),user(fk),name)

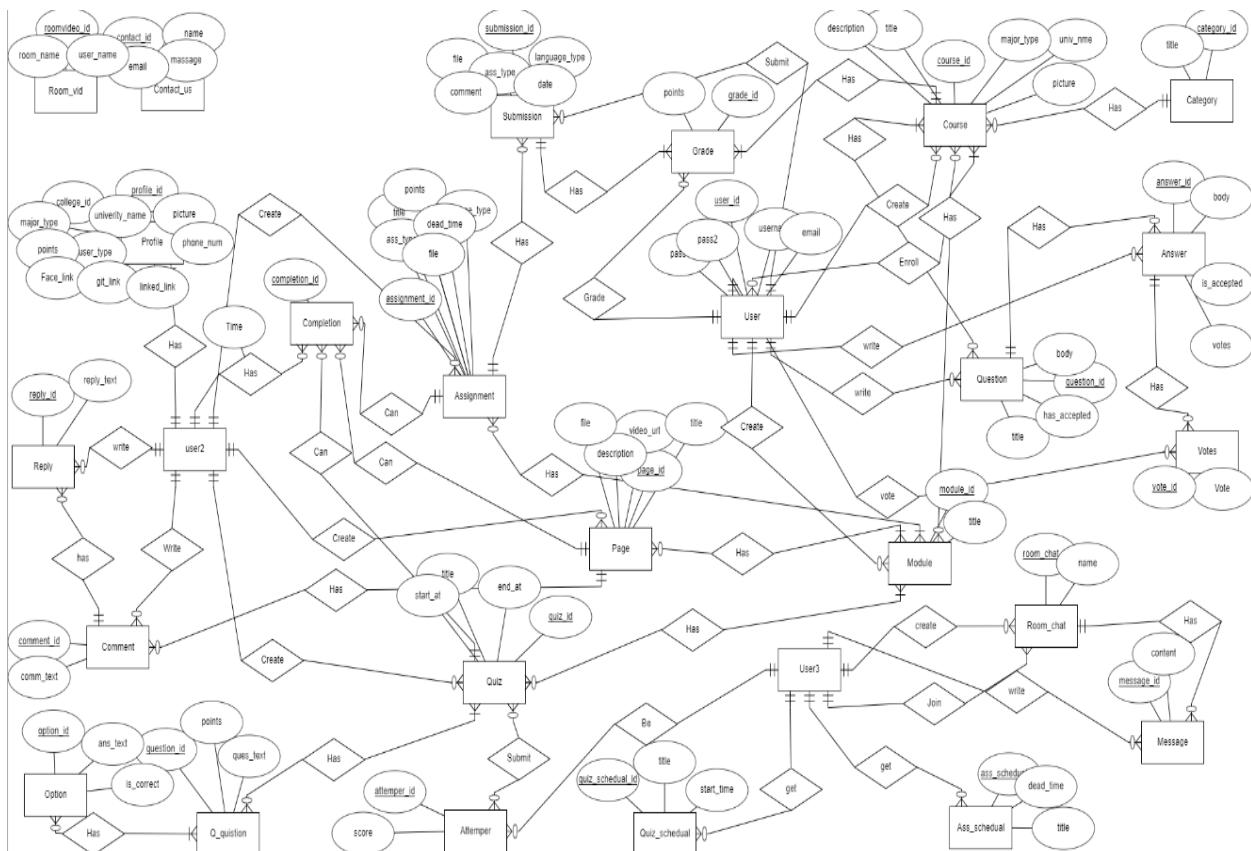
24-Message(id(pk),user(fk),room(fk),content)

25-

AssignmentScheduale(id(fk),user(fk),course(fk),module(fk),assignment(fk),title,dead_time)

26-QuizScheduale(id(pk),user(fk),course(fk),module(fk),quiz(fk),title,start_time)

The ER Diagram:



4.2.3)How did we use django to build the website?

You will find all the sourcecode in our github link:

<https://github.com/mohamed-ameer/smarterdu>

First of all we knew that django use MVT design pattern which means
1-model is the database.

2-Template is the html files.

3-View is the business logic that GET data from db or POST data into the db.it also can do some operation then show the data to the end user via the template file.

According to that in our django project we need 3 major files[models.py&views.py&template.html].

As we can use only one models.py file for all project database and we can use only one views.py file for all project functions,But it's very complicated and unorganised way to build the backend,because the project consist of many tasks like:authentication task/edit profile task/create course task ...etc.so that its recommended to divide our project into many applications(Tasks) and each application have its own models.py & views.py file, and we can collect all html files in one folder called Templates folder.

Project Structure:

-Django project:once we create a django project we can create any number of django applications,but we have only one project folder.

1-SmartEDU

-Django applications in the SmartEDU Project:

1-app_users application:for authentication tasks(login/signup/logout/forget password/edit profile).

2-classroom application:for creating course / enrolling course /show the leaderboard(which defines the most evaluated students)

3-module application:for creating a lecture/unit/topic that contains the videos,assignments and quizzes.

4-assignment application:for creating or submitting or grade assignments.

5-page application:for creating the page which contains the video of the lecture and the instructor description and comment section.

6-quiz application:for creating or submitting quizzes

7-completion:for completing(mark as done/finish like udemy website) of assignments,videos,quizzes.

8-schedule application:for sending a notification to enrolled students if the instructor has created a new assignment or quiz.

9-question application:for asking or answering questions.

10-room application:for creation or joining chat room

The Implementation of each application:

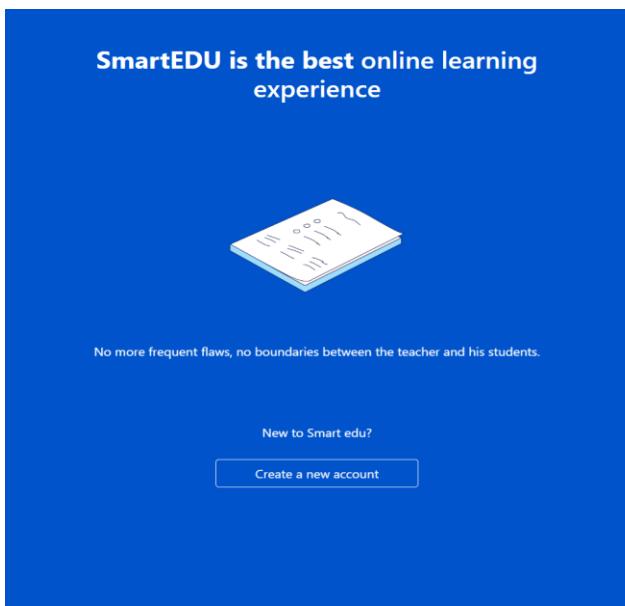
1-app_users:

1.1user:

Django has a built in user model which can simply be used in any small project and it will serve its need but this model uses username to uniquely identify a user during authentication. which can be a problem with two users having the same username here comes the need to create a custom User model but either subclassing AbstractUser or AbstractBaseUser. The first is a good option if you're happy with the existing fields on User model and just want to remove the username field, the other is starting from scratch by creating our own model, completely new User model. We choose the AbstractBaseUser to add fields we want to use in the user model.

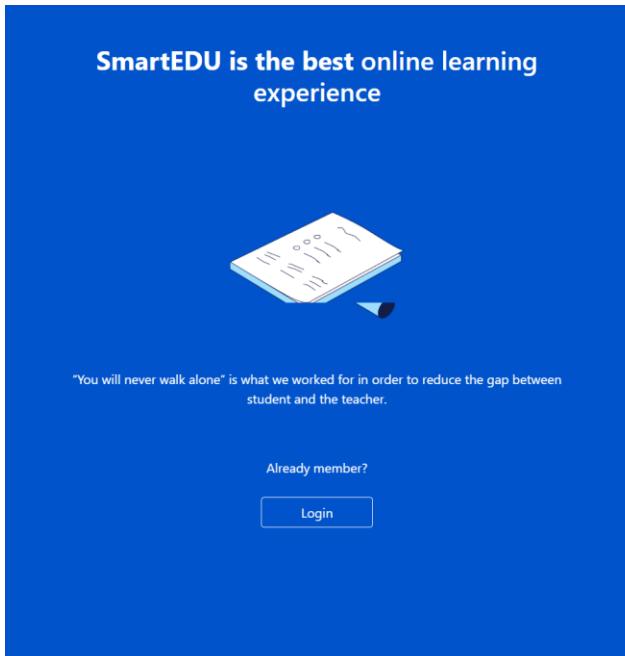
1.2login:

Django comes with a lot of built-in resources for the most common use cases of a Web application. With the Django registration app we can take advantage of some features like (Login, Logout and Password reset). First we import the - django.contrib.auth.urls- module and add a URL router for the login and logout views. this is the login form:



1.3 Register:

The most simple way to implement user sign up is by using UserCreationForm as it is. This can work when using a default Django user or with a custom user, using email to authenticate and interested only in setting username (First_Name and Last_name) and password upon sign up which looks like this:



1.4 Forget password:

Django create a unique token key for each user by using token_generator by this token we will send a validation url to the email of the user to allow him to reset his password.

SmartEDU is the best online learning experience



No more frequent flaws, no boundaries between the teacher and his students.

New to Smart edu?

[Create a new account](#)

Send Recovery Email

Email:

mohamedameer577@gmail.com

[Send](#) [Cancel](#)

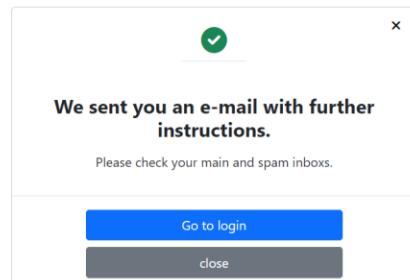
SmartEDU is the best online learning experience

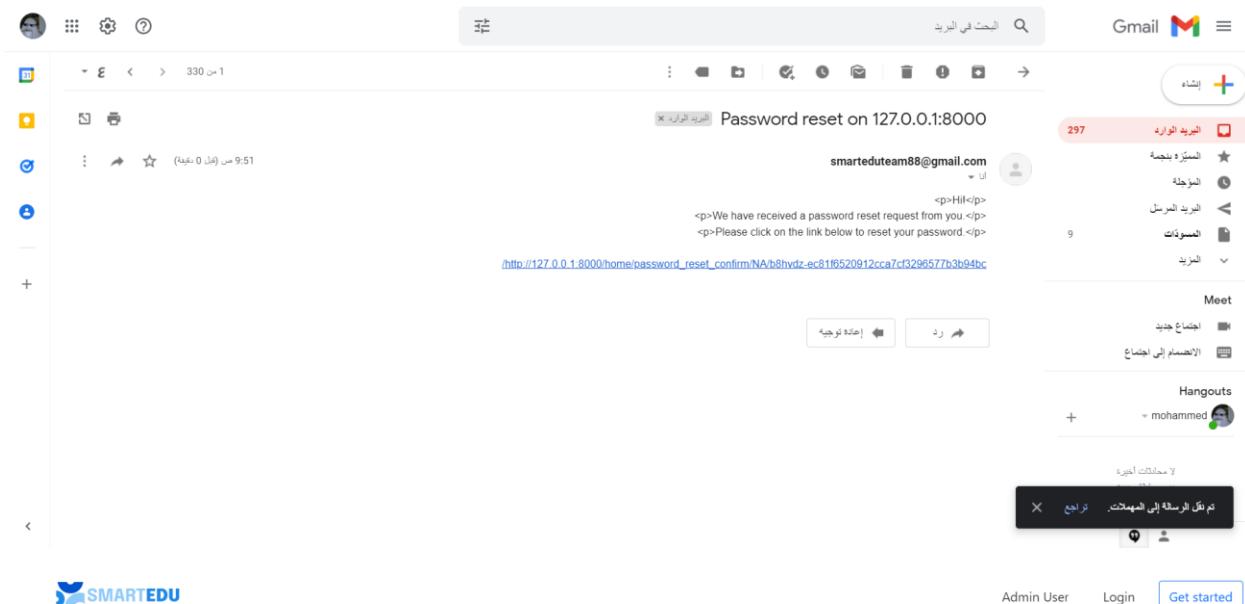


"You will never walk alone" is what we worked for in order to reduce the gap between student and the teacher.

New to Smart edu?

[Create a new account](#)





SMARTEDU

Admin User Login Get started

Reset Password

New password:

New password confirmation:

 SMARTEDU

App store

About us

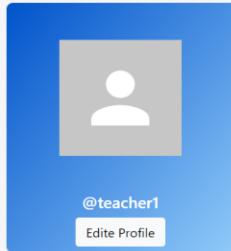


1.5Profile:

We already made a Custom user subclassing AbstractBaseUser which already helps us add more fields to the user model, but we wanted to add more fields which look more like personal things users might want to share with others to help make better communication with all users. so we created another model subclassing from the original user with oneToone relation called Profile, and by using this profile model the admin user only will decide the usertype(teacher or student)



Home Leaderboard My Courses Contact US



A blue rectangular profile card for a user named 'teacher1'. It features a placeholder profile picture, the handle '@teacher1', and a 'Edit Profile' button at the bottom right. Below the card is a large, semi-transparent light-gray rectangular overlay covering the main content area.

Information

Email teacher1@gmail.com 

Phone 01001198001 

College ID
4562123

User Type
teacher



App store

About us



Edit Profile

Personal info

Edit Profile

Username

teacher1

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

First name

First name

Last name

Last name

Email address

teacher1@gmail.com

Picture

 No file chosen

Phone

01001198001

College id

4562123

University name

Zagazig_University

Major types

electrical_engineering

Facebook

Facebook

Github

Github

Linkedin

Linkedin

Change password

 Close[App store](#)

Hii I'm a virtual assistant.
How can I help you
today?



1.6 Contact Us:

This page helps the user to contact with us by sending message or calling us

The screenshot shows the 'Contact Us' page of the Smartedu website. At the top, there is a navigation bar with links for Home, Leaderboard, My Courses, Contact US, and a user profile icon labeled 'teacher1'. Below the navigation bar is a map of the Zagazig University area. The map includes labels for Zagazig University, Faculty of Engineering Zagazig University, Zagazig Cemetery, and various local landmarks like Toba Awadha St, Health Affairs Directorate - El Sharqya, and Mr. hame. A red marker indicates the location of Zagazig University. To the right of the map is a contact form with fields for Address, Mobile, Telephone, and Email. Below these fields are input fields for Your Name, Your Email, and Your Message, along with a message box containing instructions for users. A blue 'Submit Response' button is located at the bottom of the form.



2-classroom:

According to the ERD we have created the database that allow teacher to create course with special password and allow students to see all courses then enrol this course if they have the password, the teacher can edit the course

Model class(database table)

```
50 class Course(models.Model):
51     id = models.UUIDField(primary_key=True, default=uuid.uuid4, editable=False)
52     picture = models.ImageField(upload_to=user_directory_path)
53     title = models.CharField(max_length=200)
54     secret_code = models.CharField(max_length=300, unique=True, null=True)
55     description = models.CharField(max_length=300)
56     university = models.CharField(max_length=30, choices=university, default='Z')
57     major_types = models.CharField(max_length=30, choices=major_types, default='')
58     category = models.ForeignKey(Category, on_delete=models.CASCADE)
59     # many courses created by one user(teacher)
60     user = models.ForeignKey(User, on_delete=models.CASCADE, related_name='courses')
61     # many courses can enrolled by many users(studebt)
62     enrolled = models.ManyToManyField(User)
63     modules = models.ManyToManyField(to='module.Module')
64     questions = models.ManyToManyField(Question)
65     def __str__(self):
66         return self.title
```

View functions(the operation)

```
def NewCourse(request):
    user = request.user
    if request.method == 'POST':
        form = NewCourseForm(request.POST, request.FILES)
        if form.is_valid():
            picture = form.cleaned_data.get('picture')
            title = form.cleaned_data.get('title')
            secret_code = form.cleaned_data.get('secret_code')
            description = form.cleaned_data.get('description')
            category = form.cleaned_data.get('category')
            Course.objects.create(picture=picture, title=title, secret_code=secret_code, description=description, category=category, user=user)
            return redirect('my-courses')
    else:
        form = NewCourseForm()
```

```

148 @login_required
149 def EditCourse(request, course_id):
150     user = request.user
151     course = get_object_or_404(Course, id=course_id)
152
153     if user != course.user:
154         return HttpResponseRedirect('http://127.0.0.1:8000/accounts/login')
155
156     else:
157         if request.method == 'POST':
158             form = NewCourseForm(request.POST, request.FILES, instance=course)
159             if form.is_valid():
160                 course.picture = form.cleaned_data.get('picture')
161                 course.title = form.cleaned_data.get('title')
162                 course.secret_code = form.cleaned_data.get('secret_code')
163                 course.description = form.cleaned_data.get('description')
164                 course.category = form.cleaned_data.get('category')
165                 course.save()
166             return HttpResponseRedirect('my-courses')
167
168
169 @login_required
170 def CourseDetail(request, course_id):
171     user = request.user
172     profile = Profile.objects.get(user=request.user)
173     course = get_object_or_404(Course, id=course_id)
174     context = {
175         'course': course, 'profile': profile
176     }
177
178
179     return render(request, 'classroom/course.html', context)
180
181
182 @login_required
183 def Enroll(request, course_id):
184     user = request.user
185     course = get_object_or_404(Course, id=course_id)
186     code=request.POST.get('code')
187     print(code)
188     if code == course.secret_code:
189         course.enrolled.add(user)
190         LeaderboardCourse.objects.create(user=user, course=course)
191     else:
192         return HttpResponseRedirect("Please enter the correct secret key of the course")
193     return HttpResponseRedirect('activity')

```

```

def MyCourses(request):
    user = request.user
    profile = Profile.objects.get(user=request.user)
    courses = Course.objects.filter(user=user)

    context = {
        'courses': courses, 'profile': profile
    }

    return render(request, 'classroom/mycourses.html', context)

```

The Responses

The screenshot displays the SMARTEDU platform. At the top, there is a navigation bar with links for Home, Leaderboard, My Courses, and Contact US. A user profile icon for 'teacher1' is also present.

User Profile:

- Information:**

Email	Univeristy Name
teacher1@gmail.com	Zagazig_University
- NO.Of_Courses:** 0
- major:** electrical_engineering
- Social media icons for Facebook, Twitter, and LinkedIn.
- Dr/:teacher1** (User handle)

My Courses:

- A button labeled "Add Course".
- A blue banner at the bottom left features the SMARTEDU logo and a link to the App store.
- A blue banner at the bottom right contains a message from a virtual assistant: "Hi! I'm a virtual assistant. How can I help you today?". It includes a close button and a message input field.



Create a new course

Picture

Screen Shot ...7.01 PM.png

Title

DataBase

Secret code

123

Description

this course is talking about

University

Zagazig _University

Major types

electrical_engineering

Category

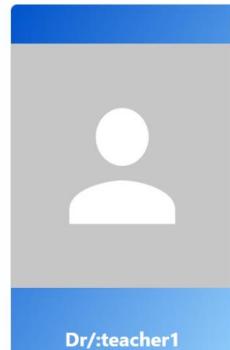
prep



App store

Hi! I'm a virtual assistant.
How can I help you
today?





Information

Email
teacher1@gmail.com 

University Name
Zagazig_University

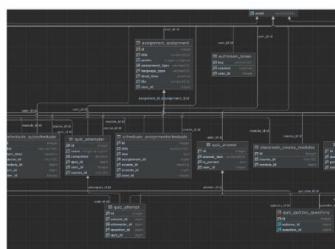
NO.Of_Courses
1

major
electrical_engineering

My Courses

Add Course



DataBase

this course is talking about

prep 0 students

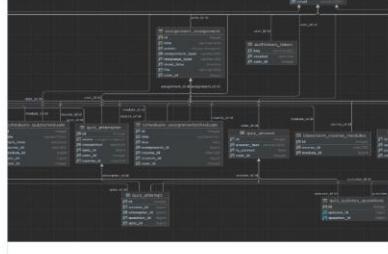
course password: 123

[EDIT COURSE or ADD MATERIAL](#)



DataBase

-  [Edit course](#)
-  [units](#)
-  [Leaderboard](#)
-  [Questions](#)
-  [Submissions](#)
-  [QuizSubmissions](#)
-  [Start Meeting](#)
-  [Chat Room](#)



DataBase

this course is talking about

prep 0 students

[Delete The Course](#)



DataBase

-  Edit course
-  units
-  Leaderboard
-  Questions
-  Submissions
-  QuizSubmissions
-  Start Meeting
-  Chat Room

Edit Course

Picture

Currently: [user_2/Screen_Shot_2022-07-11_at_8.17.01_PM.png](#)

Change: No file chosen

Title

DataBase

Secret code

123

Description

this course is talking about

University

Zagazig _University

Major types

electrical_engineering

Category

prep

 Close

Hi! I'm a virtual assistant.
How can I help you
today?





Home Leaderboard Courses Schedule Contact US

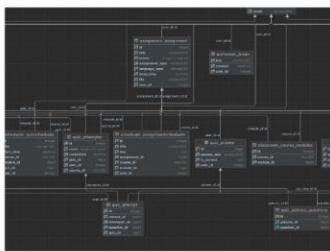
student1 ▾

Title: _____ University: _____ Major types: _____ Category: _____

Search

Advanced Search

Courses



DataBase

this course is talking about

prep 0 students

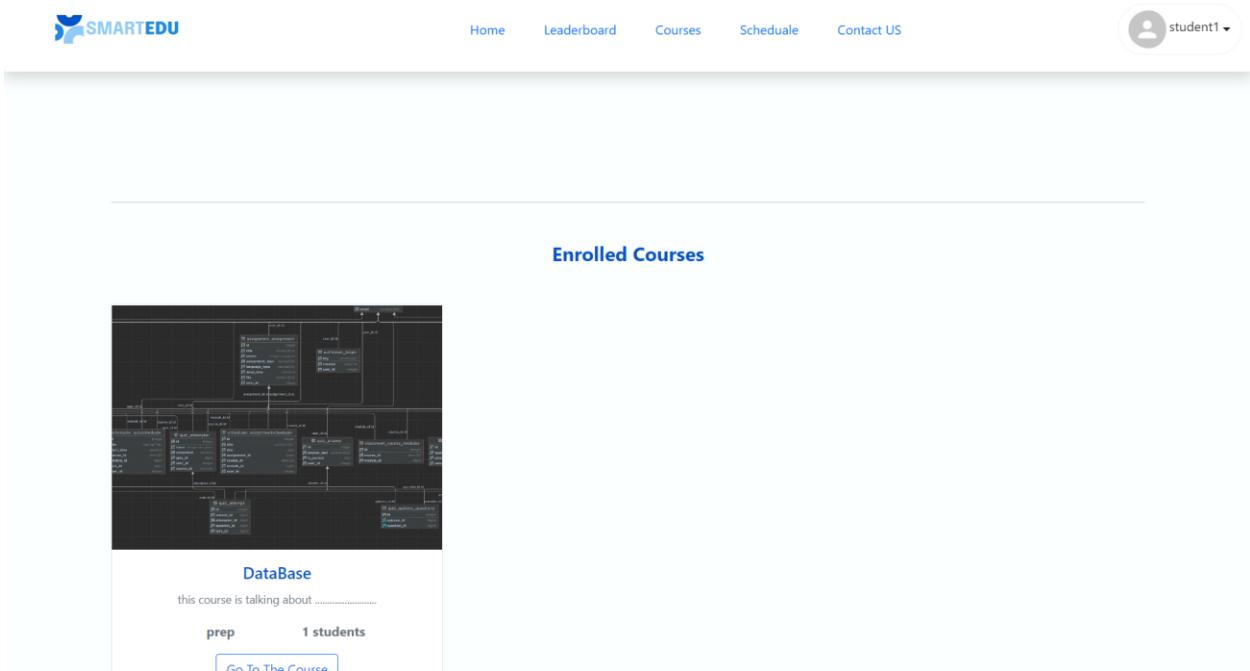
Close



App store

Hi! I'm a virtual assistant.
How can I help you
today?

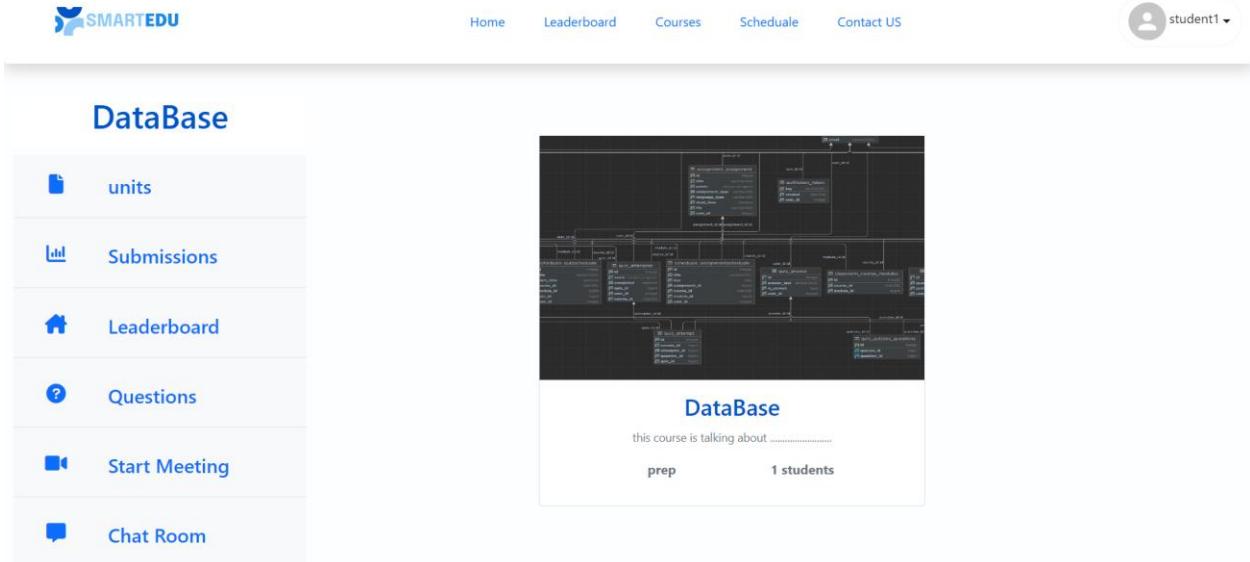




The screenshot shows the 'Enrolled Courses' section of the SmartEDU platform. At the top, there's a navigation bar with links for Home, Leaderboard, Courses, Schedule, and Contact US, along with a user profile icon for 'student1'. Below the navigation is a large, dark-themed course card for a database course. The card features a complex diagram of a database schema with various tables and their relationships. Below the diagram, the course title 'DataBase' is displayed in blue, followed by the text 'this course is talking about'. It shows 'prep' and '1 students' status, and a 'Go To The Course' button.



The screenshot shows the main home page of the SmartEDU platform. It features a prominent blue header with the 'SMARTEDU' logo on the left and an 'App store' link on the right. A blue speech bubble on the right side contains the text 'Hi! I'm a virtual assistant. How can I help you today?'. Below the header, there's a navigation bar with Home, Leaderboard, Courses, Schedule, and Contact US links, and a user profile icon for 'student1'. The main content area is currently empty, showing a light gray background.



The screenshot shows the detailed view for the 'DataBase' course. On the left, there's a sidebar with icons and links for 'units', 'Submissions', 'Leaderboard', 'Questions', 'Start Meeting', and 'Chat Room'. The main content area displays the same database schema diagram as the first screenshot. The course title 'DataBase' is shown in blue, followed by 'this course is talking about', 'prep', '1 students', and a 'Go To The Course' button.



The screenshot shows the main home page of the SmartEDU platform, similar to the previous one but with a sidebar visible on the left. The sidebar contains the same navigation links as the first screenshot: Home, Leaderboard, Courses, Schedule, and Contact US. The rest of the page is identical to the second screenshot, featuring the blue header, the 'Hi! I'm a virtual assistant...' bubble, and the empty main content area.

3-module:

For teacher we implement the database that allow him to create many modules or units in the course and in each module the teacher can create many videos or assignments or quizzes

models(the tables)

```
class Module(models.Model):
    title = models.CharField(max_length=150)
    user = models.ForeignKey(User, on_delete=models.CASCADE, related_name='modules')
    pages = models.ManyToManyField(Page)
    quizzes = models.ManyToManyField(to='quiz.Quizzes')
    quizzes_ai = models.ManyToManyField(to='quizai.QuizzesAI')
    assignments = models.ManyToManyField(to='assignment.Assignment')

    def __str__(self):
        return self.title
```

Views(the operations)

```
def NewModule(request, course_id):
    user = request.user
    course = get_object_or_404(Course, id=course_id)
    if user != course.user:
        return HttpResponseRedirect()
    else:
        if request.method == 'POST':
            form = NewModuleForm(request.POST)
            if form.is_valid():
                title = form.cleaned_data.get('title')
                m = Module.objects.create(title=title, user=user)
                course.modules.add(m)
                course.save()
                return redirect('modules', course_id=course_id)
        else:
            form = NewModuleForm()
```

```

def CourseModules(request, course_id):
    user = request.user
    profile = Profile.objects.get(user=request.user)
    course = get_object_or_404(Course, id=course_id)
    page_completions = Completion.objects.filter(user=user, course=course).values()
    quiz_completions = Completion.objects.filter(user=user, course=course).values()
    assignment_completions = Completion.objects.filter(user=user, course=course).values()
    context = {
        'profile': profile,
        'course': course,
        'page_completions': page_completions,
        'quiz_completions': quiz_completions,
        'assignment_completions': assignment_completions,
    }
    return render(request, 'module/modules.html', context)

```

The Responses:

The screenshot shows the SmartEDU platform interface. At the top, there is a navigation bar with links for Home, Leaderboard, My Courses, and Contact US. A user profile icon for 'teacher1' is also present. On the left, a sidebar titled 'DataBase' contains links for Edit course, units, Leaderboard, Questions, Submissions, QuizSubmissions, Start Meeting, and Chat Room. The main content area is titled 'Units:' and features a blue '+' button. In the bottom right corner, there is a blue chat bubble with white text that says 'Hi! I'm a virtual assistant. How can I help you today?' and a small 'Close' button.

DataBase

 Edit course units Leaderboard Questions Submissions QuizSubmissions Start Meeting Chat Room

Let's create a new module!

Title

lecture 1

[Create new module](#)[Cancel](#)

The screenshot shows the SmartEDU platform interface. At the top, there is a navigation bar with links for Home, Leaderboard, My Courses, and Contact US. A user profile icon for 'teacher1' is also present. On the left, a sidebar titled 'DataBase' contains links for Edit course, units, Leaderboard, Questions, Submissions, QuizSubmissions, Start Meeting, and Chat Room. The main content area is titled 'Units:' and shows a list with one item: '(1) lecture 1'. Below this list are four buttons: 'Add new Topic', 'Add new Quiz', 'Add new Quiz using AI', and 'Add new Assignment'. At the bottom of the page, there is a blue footer bar with the SmartEDU logo, an App store link, and a virtual assistant chat window that says 'Hi! I'm a virtual assistant. How can I help you today?'. There is also a 'Close' button in the top right corner of the chat window.

4-assignment & schedule

First for teacher we implement the database that allow him to create an assignment And for students we implement the database that send notification for student and allow them to submit the assignment then the teacher can grade their submission

Models(the tables):

```
class Assignment(models.Model):
    title = models.CharField(max_length=150)
    points = models.PositiveIntegerField()
    assignment_type = models.CharField(max_length=30, choices=assignment_type,
    language_type = models.CharField(max_length=30, choices=language_type, defa
    dead_time=models.DateTimeField()
    # files = models.ManyToManyField(AssignmentFileContent)
    file = models.FileField(upload_to=user_directory_path, blank=True, null=True)
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    def get_file_name(self):
        return os.path.basename(self.file.name)
    def __str__(self):
        return self.title
```

```

class Submission(models.Model):
    file = models.FileField(upload_to=student_directory_path)
    assignment_type = models.CharField(max_length=30, choices=assignment_type,
    language_type = models.CharField(max_length=30, choices=language_type, defa
    comment = models.CharField(max_length=1000,blank=True)
    date = models.DateTimeField(auto_now_add=True)
    assignment = models.ForeignKey(Assignment, on_delete=models.CASCADE)
    user = models.ForeignKey(User, on_delete=models.CASCADE)

    def get_file_name(self):
        return os.path.basename(self.file.name)
    def __str__(self):
        return self.assignment.title

class Grade(models.Model):
    course = models.ForeignKey(Course, on_delete=models.CASCADE)
    submission = models.ForeignKey(Submission, on_delete=models.CASCADE)
    points = models.PositiveIntegerField(default=0)
    graded_by = models.ForeignKey(User, on_delete=models.CASCADE, blank=True, n
    status = models.CharField(choices=STATUS_CHOICES, default='pending', max_le

class AssignmentScheduale(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    course=models.ForeignKey(Course, on_delete=models.CASCADE)
    module=models.ForeignKey(Module, on_delete=models.CASCADE)
    assignment=models.ForeignKey(Assignment, on_delete=models.CASCADE)
    title = models.CharField(max_length=150)
    due = models.DateField()
    def __str__(self):
        return self.title

class QuizScheduale(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    course=models.ForeignKey(Course, on_delete=models.CASCADE)
    module=models.ForeignKey(Module, on_delete=models.CASCADE)
    quiz=models.ForeignKey(Quizzes, on_delete=models.CASCADE)
    title = models.CharField(max_length=150)
    start_time=models.DateTimeField()
    def __str__(self):
        return self.title

```

Views(the operations on the tables)

```

def NewAssignment(request, course_id, module_id):
    user = request.user
    course = get_object_or_404(Course, id=course_id)
    module = get_object_or_404(Module, id=module_id)
    files_objs = []

    if user != course.user:
        return HttpResponseRedirect('forbidden')
    else:
        if request.method == 'POST':
            form = NewAssignmentForm(request.POST, request.FILES)
            if form.is_valid():
                title = form.cleaned_data.get('title')
                points = form.cleaned_data.get('points')
                dead_time = form.cleaned_data.get('dead_time')
                assignment_type = form.cleaned_data.get('assignment_type')
                language_type = form.cleaned_data.get('language_type')

#                file_instance.save()
#                files_objs.append(file_instance)
                file = request.FILES.get('file')
                a = Assignment.objects.create(title=title, points=points, file=f
# a.files.set(files_objs)
                a.save()
                module.assignments.add(a)
                AssignmentSchedule.objects.create(user=user, course=course, modu
                print('schedule object done')
                if a.assignment_type == 'Programming_File' and a.language_type
                    # marcoo functions
                    filejava=os.path.basename(file.name)#HelloWorldDoctor.java
                    filejavawithouttext=os.path.splitext(filejava)[0] #HelloWorl
                    executeJavaDoctor(filejava,filejavawithouttext)
                    print(filejavawithouttext)
return redirect('modules', course_id=course_id)

```

```

def Submissions(request, course_id):
    user = request.user
    course = get_object_or_404(Course, id=course_id)
    grades = Grade.objects.filter(course=course, submission_user=user)
    context = {
        'grades': grades,
        'course': course
    }
    return render(request, 'classroom/submissions.html', context)

def StudentSubmissions(request, course_id):
    user = request.user
    course = get_object_or_404(Course, id=course_id)
    if user != course.user:
        return HttpResponseForbidden()
    else:
        grades = Grade.objects.filter(course=course)
        myFilter=GradeFilter(request.GET,queryset=grades)
        grades =myFilter.qs
        context = {
            'course': course,
            'grades': grades,
            'myFilter': myFilter,
        }
    return render(request,'classroom/studentgrades.html', context)

def GradeSubmission(request, course_id, grade_id):
    user = request.user
    course = get_object_or_404(Course, id=course_id)
    grade = get_object_or_404(Grade, id=grade_id)

    if user != course.user:
        return HttpResponseForbidden()
    else:
        if request.method == 'POST':
            points = request.POST.get('points')
            grade.points = points
            grade.status = 'graded'
            grade.graded_by = user
            grade.save()
            LeaderboardCourse.objects.get(user=grade.submission.user, course=course).Profile.objects.get(pk=grade.submission.user.id).modify_points(int(points))
    return redirect('student-submissions', course_id=course_id)

```

The Responses:

The screenshot shows the SmartEDU platform interface. On the left, a sidebar titled "DataBase" contains links for Edit course, units, Leaderboard, Questions, Submissions, QuizSubmissions, Start Meeting, and Chat Room. The main area displays a form titled "Let's create a new Assignment!" with fields for Title (assignment1), Points (5), Assignment Type (Regular File selected), Select language (None selected), Dead time (07/13/2022 01:09 PM), and File (fromzagazig.pdf chosen). A "Create new Assignment" button is at the bottom. In the bottom right corner, a blue speech bubble from a virtual assistant says "Hi! I'm a virtual assistant. How can I help you today?".

DataBase

 Edit course units Leaderboard Questions Submissions QuizSubmissions Start Meeting Chat Room

Units:



(1) lecture 1

[Add new Topic](#) [Add new Quiz](#) [Add new Quiz using AI](#) [Add new Assignment](#)1. [assignment1](#) not Completed:  CloseHi! I'm a virtual assistant.
How can I help you today?

Schedule

Dr/teacher1 created a new assignment  assignment1

Deadtime | July 13, 2022, 1:09 p.m.

DataBase



The screenshot shows a student's assignment submission interface. On the left, a sidebar lists navigation options: units, Submissions, Leaderboard, Questions, Start Meeting, and Chat Room. The main area is titled "ASSIGNMENT" and displays the following details:

- Assignment Name: assignment1
- Submit Dead Time: July 13, 2022, 1:09 p.m.
- Assignment Type: Regular_File
- Language Type: None
- Points: 5
- File in this assignment: fromzagazig.pdf (with a download icon)
- Points: see your grades

At the bottom are two buttons: "Start Assignment" and "Cancel".

The screenshot shows the student's home page. It features the SmartEDU logo, links to App store and About us, and a user profile icon for "student1".

The screenshot shows a teacher's view of student submissions. The sidebar includes: Edit course, units, Leaderboard, Questions, Submissions, QuizSubmissions, Start Meeting, and Chat Room. The main area is titled "Submissions" and shows a table of student submissions:

Student Name	Student username	Student ID	Assignment	DeadTime	Submitted	Status	Score	Out of	Graded by	Grade
@student1	None	assignment1	assignment1	July 13, 2022, 1:09 p.m.	July 13, 2022, 9:12 a.m.	Pending	0	5	Dr/	Grade

A green "Export To Excel" button is visible above the table. A search bar and a dropdown menu are also present.

The screenshot shows the teacher's home page. It features the SmartEDU logo, links to App store and About us, and a user profile icon for "teacher1".

DataBase

-  Edit course
-  units
-  Leaderboard
-  Questions
-  Submissions
-  QuizSubmissions
-  Start Meeting
-  Chat Room

Student Name	
Student Username	@student1
Student College_ID	None
Student E_mail	student1@gmail.com
Student Phone	
Student University	Zagazig_University
Student Major	electrical_engineering
Student Assignment_File	fromzagazig.pdf 
Student comment on assignment	a

Assignment maximum points: 5

 Points 

 Close

Hi! I'm a virtual assistant.
How can I help you today?



DataBase

 Edit course units Leaderboard Questions Submissions QuizSubmissions Start Meeting Chat Room

Submissions

Open this select menu

 Export To Excel

Student Name	Student username	Student ID	Assignment	DeadTime	Submitted	Status	Score	Out of	Graded by	Grade
	@student1	None	assignment1	July 13, 2022, 1:09 p.m.	July 13, 2022, 9:12 a.m.	graded	5	5	Dr/	 Grade



App store

X Close
Hi! I'm a virtual assistant.
How can I help you today?



5-quiz & schedule:

First for teacher we implement the database that allow him to create an quiz
And for students we implement the database that sends notification for students and allows them to submit the quiz then it will grade their submission automatically.

Models(the tables)

```

class Quizzes(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    title = models.CharField(max_length=200, blank=True)
    date = models.DateTimeField(auto_now_add=True)
    start_time=models.DateTimeField()
    end_time=models.DateTimeField()
    questions = models.ManyToManyField(Question)

    def __str__(self):
        return self.title

class Answer(models.Model):
    answer_text = models.CharField(max_length=900)
    is_correct = models.BooleanField(default=False)
    user = models.ForeignKey(User, on_delete=models.CASCADE)

    def __str__(self):
        return self.answer_text

class Question(models.Model):
    question_text = models.CharField(max_length=900)
    answers = models.ManyToManyField(Answer)
    points = models.PositiveIntegerField()
    user = models.ForeignKey(User, on_delete=models.CASCADE)

    def __str__(self):
        return self.question_text

class Attempter(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    course = models.ForeignKey(to='classroom.Course', on_delete=models.CASCADE)
    quiz = models.ForeignKey(Quizzes, on_delete=models.CASCADE)
    score = models.PositiveIntegerField()
    completed = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.user.username

```

Views(the operations):

```

def NewQuiz(request, course_id, module_id):
    user = request.user
    course = get_object_or_404(Course, id=course_id)
    module = get_object_or_404(Module, id=module_id)
    if request.method == 'POST':
        form = NewQuizForm(request.POST)
        if form.is_valid():
            title = form.cleaned_data.get('title')
            start_time = form.cleaned_data.get('start_time')
            end_time = form.cleaned_data.get('end_time')
            quiz = Quizzes.objects.create(user=user, title=title, start_time=start_time, end_time=end_time)
            module.quizzes.add(quiz)
            module.save()
            QuizScheduale.objects.create(user=user, course=course, module=module, quiz=quiz)
            print('quiz scheduale done')
            return redirect('new-question', course_id=course_id, module_id=module_id)
    else:
        form = NewQuizForm()

def NewQuestion(request, course_id, module_id, quiz_id):
    user = request.user
    course = get_object_or_404(Course, id=course_id)
    quiz = get_object_or_404(Quizzes, id=quiz_id)
    if request.method == 'POST':
        form = NewQuestionForm(request.POST)
        if form.is_valid():
            question_text = form.cleaned_data.get('question_text')
            points = form.cleaned_data.get('points')
            answer_text = request.POST.getlist('answer_text')
            is_correct = request.POST.getlist('is_correct')

            question = Question.objects.create(question_text=question_text, user=user, points=points)

            for a, c in zip(answer_text, is_correct):
                answer = Answer.objects.create(answer_text=a, is_correct=c, user=user)
                question.answers.add(answer)
                question.save()
                quiz.questions.add(question)
                quiz.save()
            return redirect('new-question', course_id=course_id, module_id=module_id, quiz_id=quiz_id)
    else:
        form = NewQuestionForm()

```

```

def QuizDetail(request, course_id, module_id, quiz_id):
    user = request.user
    course = get_object_or_404(Course, id=course_id)
    module = get_object_or_404(Module, id=module_id)
    quiz = get_object_or_404(Quizzes, id=quiz_id)
    my_attempts = Attempter.objects.filter(quiz=quiz, user=user)

    context = {
        'quiz': quiz,
        'user': user,
        'my_attempts': my_attempts,
        'course_id': course_id,
        'module_id': module_id,
        'course': course,
        'module': module,
    }
    return render(request, 'quiz/quizdetail.html', context)

def TakeQuiz(request, course_id, module_id, quiz_id):
    quiz = get_object_or_404(Quizzes, id=quiz_id)
    course = get_object_or_404(Course, id=course_id)
    module = get_object_or_404(Module, id=module_id)
    context = {
        'quiz': quiz,
        'course': course,
        'module': module,
        'course_id': course_id,
        'module_id': module_id,
    }
    return render(request, 'quiz/takequiz.html', context)

```

The Responses:

DataBase

 Edit course units Leaderboard Questions Submissions QuizSubmissions Start Meeting Chat Room

Let's create a new Quiz!

Title

quiz

Start time

07/13/2022 11:33 AM



End time

07/13/2022 12:33 PM

[Create new Quiz](#)[Cancel](#)

DataBase

 Edit course units Leaderboard Questions Submissions QuizSubmissions Start Meeting Chat Room

Let's create a new Question!

Question text

question1

Points

Points

answer a Answer 1

Yes Is this the correct answer ?

answer b Answer 2

No Is this the correct answer ?

answer c Answer 3

No Is this the correct answer ?

answer d Answer 4

No Is this the correct answer ?[Save this question and Add another one](#) [Finish](#)[App store](#)

X Close
Hi! I'm a virtual assistant.
How can I help you
today?



Schedule

Dr/teacher1 created a new assignment  assignment1

 Deadtime | July 13, 2022, 1:09 p.m.

DataBase

Dr/teacher1 created a new Quiz  quiz1

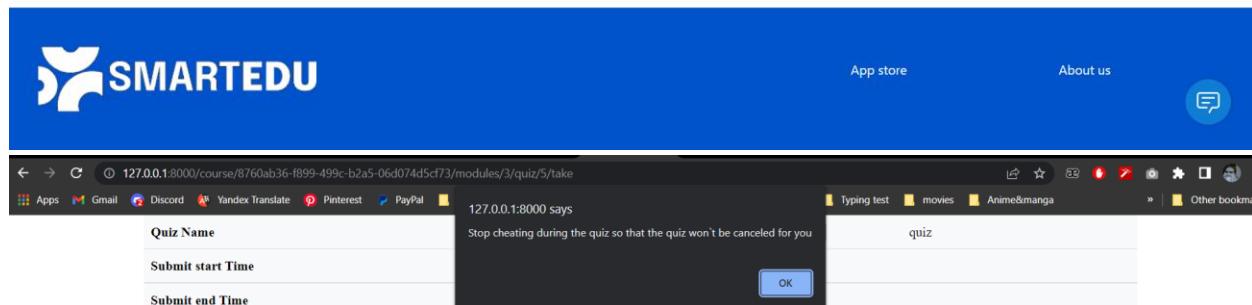
 Start time | July 13, 2022, 11:32 a.m.

DataBase

Dr/teacher1 created a new Quiz  quiz

 Start time | July 13, 2022, 11:33 a.m.

DataBase



The screenshot shows a web browser window with the SMARTEDU logo at the top. The address bar indicates the URL is 127.0.0.1:8000/course/8760ab36-f899-499c-b2a5-06d074d5cd73/modules/3/quiz/5/take. A modal dialog box is centered on the screen, displaying the message "Stop cheating during the quiz so that the quiz won't be canceled for you" with an "OK" button. The main page content includes fields for "Quiz Name", "Submit start Time", and "Submit end Time". On the right side, there is a sidebar with links like "App store", "About us", and a message icon. The bottom of the page shows a navigation bar with links for "Typing test", "movies", "Anime&manga", and "Other bookmarks".

1 Question

*Choose the correct answer for all questions

56m :30s

(1) question1

- answer a
- answer b
- answer c
- answer d

[Submit](#)

DataBase

QUIZ

SUBMITTED

-  units
-  Submissions
-  Leaderboard
-  Questions
-  Start Meeting
-  Chat Room

Quiz Name	quiz
Submit Start Time	July 13, 2022, 11:33 a.m.
Submit End Time	July 13, 2022, 12:33 p.m.
student1 score	Scored: 2

[Cancel](#)[App store](#)

Hi! I'm a virtual assistant.
How can I help you
today?



Quiz Name

quiz

Submit Start Time

July 13, 2022, 11:33 a.m.

Submit End Time

July 13, 2022, 12:33 p.m.

student1 score

Scored: 2

1 Question

*Choose the correct answer for all questions

(1)question1

- answer a
 answer b
 answer c
 answer d

Your answers

answer a

[App store](#)

Hi! I'm a virtual assistant.
How can I help you
today?



DataBase

-  Edit course
-  units
-  Leaderboard
-  Questions
-  Submissions
-  QuizSubmissions
-  Start Meeting
-  Chat Room

QuizSubmissions

[Export To Excel](#)

Student Name	Student username	Student ID	Quiz	score
	@student1	None	quiz	2



6-page & discussion

for teacher we implement the database that allow him to create an video section
And for students we implement the function that allow them to complete the video
and discuss about the video content.

Models(the table)

```

class Page(models.Model):
    title = models.CharField(max_length=150)
    video_url = EmbedVideoField()
    description = models.CharField(max_length=300)
    # files = models.ManyToManyField(PostFileContent)
    file = models.FileField(upload_to=user_directory_path, blank=False)
    user = models.ForeignKey(User, on_delete=models.CASCADE, related_name='page_owner')
    def get_file_name(self):
        return os.path.basename(self.file.name)
    def __str__(self):
        return self.title

class Comment(models.Model):
    user=models.ForeignKey(User,on_delete=models.CASCADE)
    page=models.ForeignKey(Page,on_delete=models.CASCADE)
    comm=models.TextField()
    time=models.DateTimeField(auto_now_add=True,auto_now=False)
    def __str__(self):
        return self.comm
class Reply(models.Model):
    user=models.ForeignKey(User,on_delete=models.CASCADE)
    page=models.ForeignKey(Page,on_delete=models.CASCADE,null=True)
    comm=models.TextField()
    time=models.DateTimeField(auto_now_add=True,auto_now=False)

```

Views(the operations)

```

@login_required
def NewPageModule(request, course_id, module_id):
    user = request.user
    course = get_object_or_404(Course, id=course_id)
    module = get_object_or_404(Module, id=module_id)
    files_objs = []

    if user != course.user:
        return HttpResponseForbidden()

    else:
        if request.method == 'POST':
            form = NewPageForm(request.POST, request.FILES)
            if form.is_valid():
                title = form.cleaned_data.get('title')
                video_url = form.cleaned_data.get('video_url')
                description = form.cleaned_data.get('description')
                file = request.FILES.get('file')
                p.save()
                module.pages.add(p)
                return redirect('modules', course_id=course_id)
        else:
            form = NewPageForm()
    context = {
        'form': form, 'course': course
    }

    return render(request, 'page/newpage.html', context)

```

```

def PageDetail(request, course_id, module_id, page_id):
    user=request.user
    page = get_object_or_404(Page, id=page_id)
    course = get_object_or_404(Course, id=course_id)
    module = get_object_or_404(Module, id=module_id)
    completed = Completion.objects.filter(course_id=course_id, user=request.user, page_id=page_id)

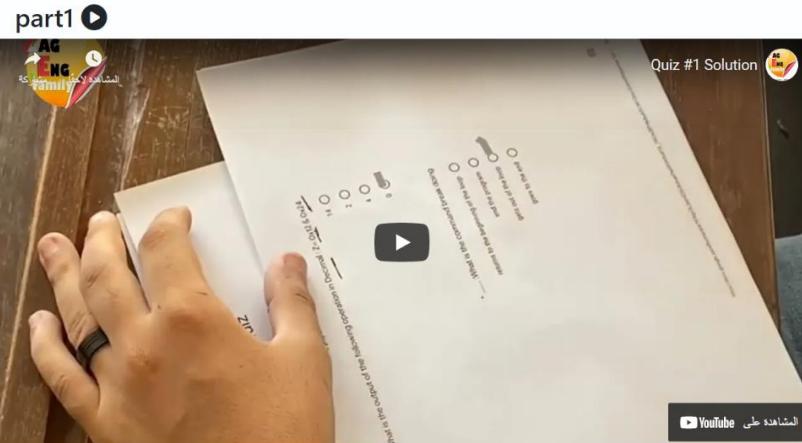
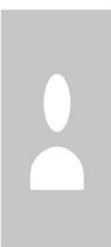
    if request.method == 'POST':
        comm = request.POST.get('comm')
        comm_id = request.POST.get('comm_id') #None

        if comm_id:
            Reply(page=page,
                  user = request.user,
                  comm = comm,
                  comment= Comment.objects.get(id=int(comm_id))
                  ).save()
        else:
            Comment(page=page, user=request.user, comm=comm).save()
    points = 1
    Profile.objects.get(pk=user.id).modify_points(points)

```

The Responses:

The screenshot shows the SmartEDU platform interface. On the left, there's a sidebar with navigation links: Edit course, units, Leaderboard, Questions, Submissions, QuizSubmissions, Start Meeting, and Chat Room. The main content area has a title "Let's create a new Topic!" and fields for Title (part1), Video url (https://www.youtube.com/watch?v=Xu2_8pDHYTA), Description (in this part of lecture you have to.....), and File (Choose File fromzagazig.pdf). At the bottom right of the main area, there's a blue callout bubble with white text: "Hi! I'm a virtual assistant. How can I help you today?". The footer features the SmartEDU logo and links to the App store and Google Play.

Attachments  [fromzagazig.pdf](#) Professor description  teacher1

in this part of lecture you have to.....

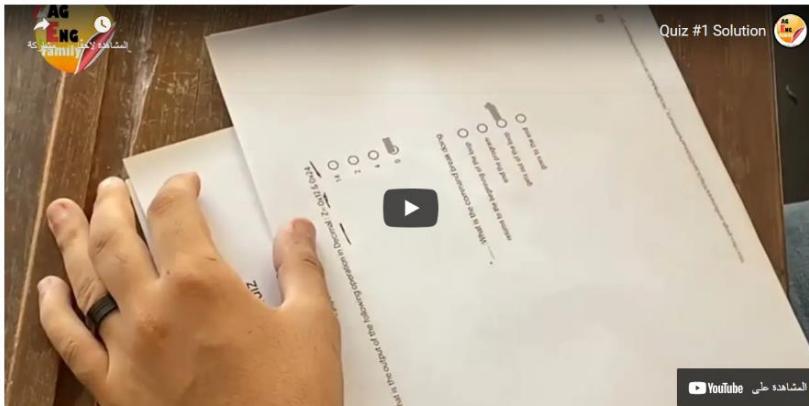
 [Mark as done](#)[Edit Page](#)Discussion 

Comments

Leave a comment

Comment

 Post Comment Close

part1 Attachments  [fromzagazig.pdf](#) Professor description 

teacher1

in this part of lecture you have to.....

 [Mark as done](#)Discussion 

Comments

Leave a comment

Comment

انا مش فاهم ليه طقططقطقطط؟



student1

today

انا مش فاهم ليه طقططقطقطط؟

 Hi! I'm a virtual assistant.
How can I help you today?

7-question & answer section

For both teacher and students we implement the database that allows them to create questions or answers and allow the user who creates the question and the course's teacher to mark certain answers as correct answers.

Models(the tables)

```
class Question(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE, related_name='ques
    title = models.CharField(max_length=300)
    body = RichTextField()
    created_date = models.DateTimeField(auto_now_add=True)
    update_date = models.DateTimeField(auto_now=True)
    has_accepted_answer = models.BooleanField(default=False)

    def __str__(self):
        return self.title

    def get_answers_count(self):
        return Answer.objects.filter(question=self).count()

class Answer(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE, related_name='answ
    question = models.ForeignKey(Question, on_delete=models.CASCADE)
    body = RichTextField()
    created_date = models.DateTimeField(auto_now_add=True)
    update_date = models.DateTimeField(auto_now=True)
    votes = models.IntegerField(default=0)
    is_accepted_answer = models.BooleanField(default=False)

    def __str__(self):
        return self.user.username

    def calculate_votes(self):
        u_votes = Votes.objects.filter(answer=self, vote='U').count()
        d_votes = Votes.objects.filter(answer=self, vote='D').count()
        self.votes = u_votes - d_votes
        self.save()
        return self.votes
```

```

class Votes(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE, related_name='votes')
    answer = models.ForeignKey(Answer, on_delete=models.CASCADE, related_name='votes')
    vote = models.CharField(choices=VOTES_CHOICES, max_length=1)
    date = models.DateTimeField(auto_now_add=True)

```

Views(the operations)

```

def NewStudentQuestion(request, course_id):
    user = request.user
    course = get_object_or_404(Course, id=course_id)

    if request.method == 'POST':
        form = QuestionForm(request.POST)
        if form.is_valid():
            title = form.cleaned_data.get('title')
            body = form.cleaned_data.get('body')
            q = Question.objects.create(user=user, title=title, body=body)
            course.questions.add(q)
            return redirect('questions', course_id=course.id)
    else:
        form = QuestionForm()
    context = {
        'form': form, 'course': course
    }
    return render(request, 'question/newquestion.html', context)

def Questions(request, course_id):
    course = get_object_or_404(Course, id=course_id)
    questions = course.questions.all()

    #Pagination
    paginator = Paginator(questions, 10)
    page_number = request.GET.get('page')
    questions_data = paginator.get_page(page_number)

    context = {
        'course': course,
        'questions': questions_data,
    }
    return render(request, 'question/questions.html', context)

```

```

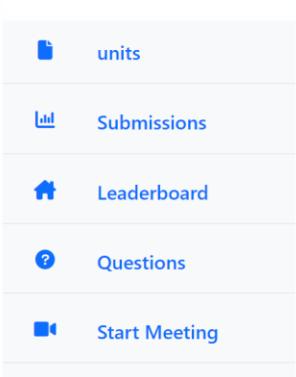
def MarkAsAnswer(request, course_id, question_id, answer_id):
    user = request.user
    course = get_object_or_404(Course, id=course_id)
    question = get_object_or_404(Question, id=question_id)

    if user == course.user or user == question.user:
        answer = get_object_or_404(Answer, id=answer_id)
        answer.is_accepted_answer = True
        answer.save()
        question.has_accepted_answer = True
        question.save()
        points = 1
        Profile.objects.get(pk=answer.user.id).modify_points(points)
        print('done')
        return redirect('question-detail', course_id=course_id, question_id=question_id)
    else:
        return HttpResponseForbidden()

def VoteAnswer(request, course_id, question_id):
    user = request.user
    answer_id = request.POST['answer_id']
    vote_type = request.POST['vote_type']
    try:
        answer = get_object_or_404(Answer, id=answer_id)
        voted = Votes.objects.filter(user=user, answer=answer)
        if voted:
            voted.delete()
        else:
            if vote_type == 'U':
                Votes.objects.create(user=user, answer=answer, vote='U')
            else:
                Votes.objects.create(user=user, answer=answer, vote='D')
    except Exception as e:
        raise e

```

The Responses:

[Home](#) [Leaderboard](#) [Courses](#) [Schedule](#) [Contact US](#) student1

DataBase

-  units
-  Submissions
-  Leaderboard
-  Questions
-  Start Meeting
-  Chat Room

Add Question: 

The Questions:

The image shows a screenshot of the SmartEdu application's user interface. At the top, there is a blue header bar with the 'SMARTEDU' logo on the left, and 'App store' and 'About us' links on the right. A circular profile icon with a person icon and the text 'student1' is also in the top right corner. Below the header, the main content area has a white background. On the left, there is a vertical sidebar with icons and labels: 'units' (blue square), 'Submissions' (blue line chart), 'Leaderboard' (blue house), 'Questions' (blue question mark), 'Start Meeting' (blue video camera), and 'Chat Room' (blue speech bubble). The main content area features a title 'Ask Your Question!' in blue, followed by a 'Question' section. Inside this section, there is a text input field containing 'whyaaaaaaaaaa?' and a 'Title' label next to it. Below the input field is a rich text editor toolbar with various styling options like bold, italic, underline, and font sizes. Underneath the toolbar is a text area labeled 'question section' with placeholder text 'question section'. At the bottom of the page, there are two buttons: a blue 'Create New Question' button and a white 'Cancel' button.

 SMARTEDU

App store

Hil I'm a virtual assistant.
How can I help you
today?

SMARTEDU

Home Leaderboard Courses Schedule Contact US

 student1 +

 SMARTEDU

[App store](#)

Hil I'm a virtual assistant.
How can I help you today?

 SMARTEDU

[App store](#)

Hil I'm a virtual assistant.
How can I help you
today?

8-Chat Room(this feature will be improved in next version)

for teacher we implement the database that allow him to create an chat room
And for students we implement the function that allow them to join the room

Models(the table)

```
class Room(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE, related_name='room')
    course = models.ForeignKey(to='classroom.Course', on_delete=models.CASCADE)
    name = models.CharField(max_length=255, unique=True)
    slug = models.SlugField(null=True, blank=True)
    def __str__(self):
        return self.name

    def save(self, *args, **kwargs):
        self.slug = slugify(self.name)
        super().save(*args, **kwargs)

class Message(models.Model):
    room=models.ForeignKey(Room,related_name='messages',on_delete=models.CASCADE)
    user=models.ForeignKey(User,related_name='messages',on_delete=models.CASCADE)
    content =models.TextField()
    date_added=models.DateTimeField(auto_now_add=True)
```

Views(the operations)

```

def rooms(request, course_id):
    course = get_object_or_404(Course, id=course_id)
    rooms = Room.objects.filter(course=course)
    myFilter=RoomFilter(request.GET,queryset=rooms)
    rooms =myFilter.qs
    return render(request, 'room/rooms.html', {'rooms': rooms,'course':course,'

@login_required
def room(request, slug):
    room = Room.objects.get(slug=slug)
    messages=Message.objects.filter(room=room)[0:25]
    return render(request, 'room/room.html', {'room': room,'messages': messages

def NewRoom(request, course_id):
    user = request.user
    course = get_object_or_404(Course, id=course_id)
    if request.method == 'POST':
        form = RoomForm(request.POST)
        if form.is_valid():
            name = form.cleaned_data.get('name')
            Room.objects.create(name=name,user=user,course=course)
            return redirect('rooms', course_id=course_id)
    else:
        form = RoomForm()

    context = {
        'form': form,'course':course
    }

    return render(request, 'room/newroom.html', context)

```

The Responses:



Home Leaderboard My Courses Contact US



DataBase

- [Edit course](#)
- [units](#)
- [Leaderboard](#)
- [Questions](#)
- [Submissions](#)
- [QuizSubmissions](#)
- [Start Meeting](#)
- [Chat Room](#)

Search By Room Name

search

Create NewRoom



Home Leaderboard My Courses Contact US



DataBase

Add Room

Name

[Save](#) [Cancel](#)



App store

X Close
Hi! I'm a virtual assistant.
How can I help you today?





Home Leaderboard My Courses Contact US

teacher1 ▾

DataBase

Edit course

units

Leaderboard

Questions

Submissions

QuizSubmissions

Start Meeting

Chat Room

Search By Room Name

search

section 1&2

this room is created
by:@teacher1

Join

section 3&4

this room is created
by:@teacher1

Join

Create NewRoom



Home Leaderboard Courses Schedule Contact US

student1 ▾

DataBase

units

Submissions

Leaderboard

Questions

Start Meeting

Chat Room

Search By Room Name

search

section 1&2

this room is created
by:@teacher1

Join

section 3&4

this room is created
by:@teacher1

Join

midterm revision

this room is created
by:@teacher1

Join

section 1&2

this room is created by:@teacher1



8-Chat video(this feature will be improved in next version)

for teacher we implement the database that allow him to create an chat video

And for students we implement the function that allow them to join the video chat

We added this feature to test it but in next version we will make it as similar as zoom meeting, so follow our github update :

<https://github.com/mohamed-ameer/smarterdu>

Models(the tables)we make the database public until now

```
class RoomMember(models.Model):
    name = models.CharField(max_length=200)
    uid = models.CharField(max_length=1000)
    room_name = models.CharField(max_length=200)
    insession = models.BooleanField(default=True)

    def __str__(self):
        return self.name
```

Views(the operations)

```

def getToken(request):
    appId = "11472a6f798747d0b6b3f50736941c6d" #YOUR APP ID
    appCertificate = "3f0a4e5d137e4ee488073d4f4c9e2a7d" #YOUR APP CERTIFICATE
    channelName = request.GET.get('channel')
    uid = random.randint(1, 230)
    expirationTimeInSeconds = 3600
    currentTimeStamp = int(time.time())
    privilegeExpiredTs = currentTimeStamp + expirationTimeInSeconds
    role = 1

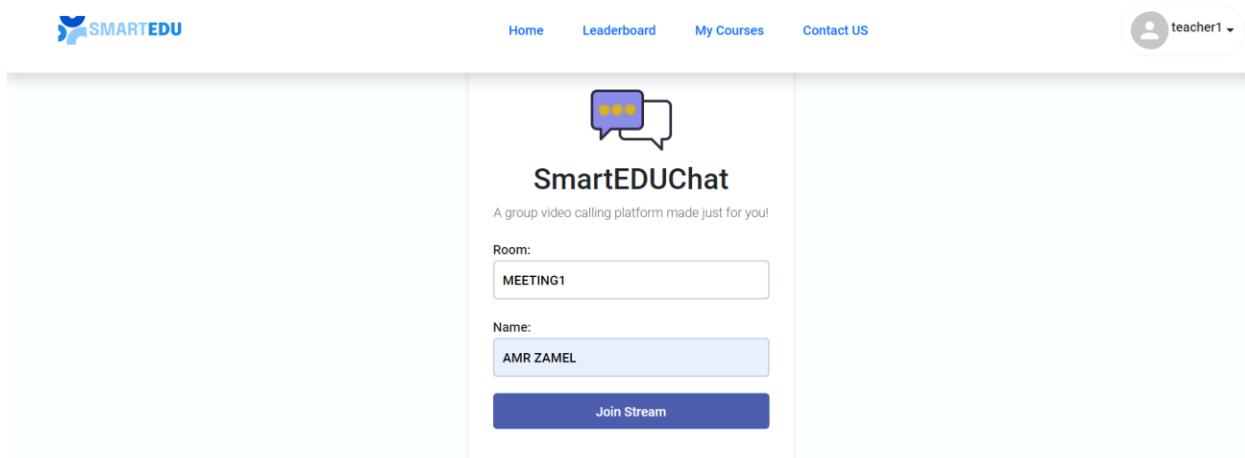
    token = RtcTokenBuilder.buildTokenWithUid(appId, appCertificate, channelNam
    return JsonResponse({'token': token, 'uid': uid}, safe=False)

@csrf_exempt
def createMember(request):
    data = json.loads(request.body)
    member, created = RoomMember.objects.get_or_create(
        name=data['name'],
        uid=data['UID'],
        room_name=data['room_name']
    )

    return JsonResponse({'name': data['name']}, safe=False)

```

The Responses:





SMARTEDU

Home Leaderboard My Courses Contact US

teacher1

section 1&2

this room is created by:@teacher1

@student1 السلام عليكم حد عارف دددددددددددددد today

@teacher1 هندياء ميتيج وده اسم الميتيج today

Your message...if you want to send emoji 😊 click[windows key + semicolon key();] Send ➔

SMARTEDU

Home Leaderboard Courses Schedule Contact US

student1

section 1&2

this room is created by:@teacher1

@student1 السلام عليكم حد عارف دددددددددددددد today

@teacher1 هندياء ميتيج وده اسم الميتيج today

Your message...if you want to send emoji 😊 click[windows key + semicolon key();] Send ➔

The image shows a screenshot of the SmartEDUChat application interface. At the top, there is a navigation bar with the SmartEDU logo, Home, Leaderboard, Courses, Schedule, Contact US, and a user profile icon labeled "student1". Below the navigation bar is a central panel titled "SmartEDUChat" with a speech bubble icon. It displays a message: "A group video calling platform made just for you!". It includes input fields for "Room" (containing "MEETING1") and "Name" (containing "MOHAMED"), followed by a blue "Join Stream" button. To the left of the main panel, there is a dark overlay with permission requests: "Use your camera" (with a camera icon) and "Use your microphone" (with a microphone icon). Below these are "Allow" and "Block" buttons. The main video area shows a participant named "amr zamel" on the left and "mohamed" on the right. At the bottom of the video area are three red control icons: a microphone, a video camera, and a hand. The overall background is light gray.

In next version we will try to improve this features

9-Leaderboard:

We have created a table that calculates the grade of students for assignments and quizzes and shows it as a leaderboard.

Each course will have its own leaderboard

Models

```
class LeaderboardCourse(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE, related_name='lead
    course = models.ForeignKey(Course, on_delete=models.CASCADE)
    points = models.PositiveIntegerField(default=0, verbose_name="points")
    def modify_points(self, added_points):
        self.points += added_points
        self.save()
    def __str__(self):
        return self.user.username
```

Views

```
def Leaderboardeachcourse(request, course_id):
    course = get_object_or_404(Course, id=course_id)
    leaders = LeaderboardCourse.objects.all().filter(course=course).order_by("-p
    context = {
        'leaders': leaders, 'course': course
    }
    return render(request, 'classroom/Leaderboardcourse.html', context)
```

Responses points = assignments + quizzes grade that the student get in this course

The image shows a screenshot of the SmartEDU application. On the left, there is a sidebar titled "DataBase" with the following options:

- Edit course
- units
- Leaderboard
- Questions
- Submissions
- QuizSubmissions
- Start Meeting
- Chat Room

On the right, there is a "Leaderboard" section with a table:

	@Username	Full Name	Student ID	Points
1	@student1		None	7

How did we build the API to be used by the mobile application ?

You can see our API via this link:

<https://documenter.getpostman.com/view/19584619/UzJFwJhM>

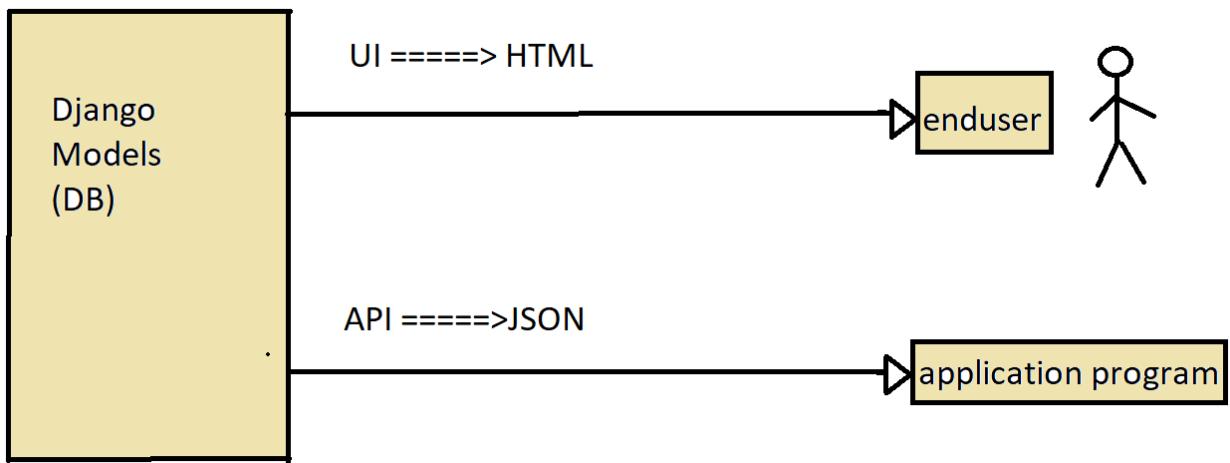
An application programming interface is a connection between computers or between computer programs. It is a type of software interface, offering a service to other pieces of software. A document or standard that describes how to build such a connection or interface is called an API specification we used rest framework to provide the project with api protocols to use rest api

Add 'rest_framework' to your INSTALLED_APPS setting.

```
>> INSTALLED_APPS = [ 'rest_framework', ]
```

Then whenever we want to use any functions provided by rest framework we just

```
>> from rest_framework import routers, serializers, viewsets
```



In short, what is API?

-If we want to post or get data to/from the end user(human/not program),then we will send or get the data in user interface (UI) response which is nothing but HTML.

-If we want to post or get data to/from application program(mobile app/java app/python app/not user/not human),then we will send or get the data in Application Programming Interface(API) response which is nothing but JSON.

*Before start explaining how we built the API,we have to know Serializer concept:
 1-Serializer is kind of a converter that converts the complex type of data to a native python data that can be rendered as JSON or XML which our frontend can understand.

2-Django models include complex data,so we need to use ModelSerializer to deal with it.

3-all we have to do in our serializer is to tell it:

- a)which model that we need to serialise data from
- b)which fields that we need our serializer to handle

*To create a new API project, first set up a Python environment in our working directory. And after that we installed Django and Django REST framework into the virtual environment. we added a folder called api into our APPs. And after that created 3 file views.py,serializer.py and urls.py.

1-User Authentication:

1.1use:

```
10 class ProfileSerializer(serializers.ModelSerializer):
11     class Meta:
12         model=Profile
13         fields=['user_type',]
14
15 class UserSerializer(serializers.ModelSerializer):
16     profile = ProfileSerializer(write_only=True)
17     password2=serializers.CharField(style={"input_type":"password"}, write_only=True)
18     class Meta:
19         model=User
20         fields=['username', 'email', 'password', 'password2','profile']
21     def save(self, **kwargs):
22         user=User(
23             username=self.validated_data['username'],
24             email=self.validated_data['email']
25         )
26         password=self.validated_data['password']
27         password2=self.validated_data['password2']
28         if password !=password2:
29             raise serializers.ValidationError({"error":"password do not match"})
30         user.set_password(password)
31         user.save()
32         return user
```

In the figure shown above, we created the serializer which will get the data and parse it .then we defined the attributes with a quick way by setting the **model = User** and then we add the extra field we want which is password2 and usertype by setting profile = ProfileSerializer().

```
12 class SignupView(generics.GenericAPIView):
13     serializer_class=UserSerializer
14     permission_classes = [AllowAny]
15     def post(self, request, *args, **kwargs):
16         serializer=self.get_serializer(data=request.data)
17         serializer.is_valid(raise_exception=True)
18         user=serializer.save()
19         return Response({
20             "user":UserSerializer(user, context=self.get_serializer_context()),
21             "profile":Profile.objects.get(user=user).user_type,
22             "token":Token.objects.get(user=user).key,
23             "message":"account created successfully"
24         })
```

In the views function, if the data is valid then we execute the save method and create the account. We check the passwords if they are equal then >> account.set_password(password) and finally we save the account instance and return the account info and the token of the user.

1.2Sign In

first we need to define an API token which is a unique identifier of an application requesting access to your service. Your service would generate an API token for the application to use when requesting your service. You can then match the token they provide to the one you store in order to authenticate,each user has his/her unique token key.

```
class CustomAuthToken(ObtainAuthToken):
    def post(self, request, *args, **kwargs):
        serializer=self.serializer_class(data=request.data, context={'request': request})
        serializer.is_valid(raise_exception=True)
        user=serializer.validated_data['user']
        token, created=Token.objects.get_or_create(user=user)
        profile, created=Profile.objects.get_or_create(user=user)
        return Response({
            'token':token.key,
            'user_id':user.pk,
            'profile':profile.user_type
        })
```

if the method was post method then the post function is being run if the data is valid then create or get the token and return the following response token : so the user can send requests using it and use it to navigate the server.

Authentication

POST Register/signup

```
http://127.0.0.1:8000/api/app_users/signup/
```

Add a user to our database and determined the usertype either "student" user or "teacher" user

BODY raw

```
{
    "username": "student15",
    "email": "student15@gmail.com",
    "password": "asdfzxcv123",
    "password2": "asdfzxcv123",
    "profile": {"user_type": "student"}
}
```

POST Login user

```
http://127.0.0.1:8000/api/app_users/login/
```

BODY raw

```
{  
    "username": "student15",  
    "password": "asdfzxcv123"  
}
```

POST logout

```
http://127.0.0.1:8000/api/app_users/logout/
```

HEADERS

Authorization token e1c0dd769309857d4ee4706a58d477f66cdf12ab

1.3-Profile:

In the database we set the profile to be created automatically whenever an account is created thus we just implement a function for editing the profile serializer as we did before we set the model to be profile and the data field to be first and last name and phone number and profile picture and save method is executed whenever it's been called after validation the data.

```
class GetProfileSerializer(serializers.ModelSerializer):  
    class Meta:  
        model=Profile  
        fields='__all__'  
  
# authentication  
class ProfileSerializer(serializers.ModelSerializer):  
    class Meta:  
        model=Profile  
        fields=[ 'user_type', ]
```

we set the permission to isAuthenticated only or read only then whenever a request came with the header we extract the token from the request then update the profile and if the token is raise an exception and return invalid token url.

```
47 class ProfileViewSet(viewsets.ModelViewSet):  
48     queryset = Profile.objects.all()  
49     serializer_class = GetProfileSerializer  
50
```

User Profile

GET all users profile

```
http://127.0.0.1:8000/api/app_users/profiles/
```

you can get all user profiles in the database and you can get a specific user profile with the ID of the user then you can provide {get,put,delete } methods

HEADERS

```
Authorization token 80f4edf98e66878a25424f77fa245c1ecc146302
```

2-Classroom:

Like we did before to communicate with classroom models we need to 1-define in our serializer:

- a)which model that we need to serialise data from
- b)which fields that we need our serializer to handle

```
4 class CategorySerializer(serializers.ModelSerializer):  
5     class Meta:  
6         model=Category  
7         fields='__all__'  
8 # get put update delete Course data  
9 class CourseSerializer(serializers.ModelSerializer):  
10    class Meta:  
11        model=Course  
12        fields='__all__'  
13 # get put update delete Grade data  
14 class GradeSerializer(serializers.ModelSerializer):  
15    class Meta:  
16        model=Grade  
17        fields='__all__'
```

2-inherit ModelViewSet class in our View Classes, and in each class we need to define in them

a) which serializer we need to use

```
5 class CategoryViewSet(viewsets.ModelViewSet):
6     queryset = Category.objects.all()
7     serializer_class = CategorySerializer
8
9 class CourseViewSet(viewsets.ModelViewSet):
10    queryset = Course.objects.all()
11    serializer_class = GradeViewSet
12 class GradeViewSet(viewsets.ModelViewSet):
13     queryset = Grade.objects.all()
14     serializer_class = GradeSerializer
```

3-that is the url that the mobile will use:

classroom

here you can create courses and category and grade and you can use all http methods
(get-post-put-delete)with these apis

GET create delete update get courses

http://127.0.0.1:8000/api/classroom/course/

HEADERS

Authorization token 80f4edf98e66878a25424f77fa245c1ecc146302

Example Request

create delete update get courses

```
curl --location --request GET 'http://127.0.0.1:8000/api/classroom/course/' \
--header 'Authorization: token 80f4edf98e66878a25424f77fa245c1ecc146302'
```

GET create delete update get category

http://127.0.0.1:8000/api/classroom/category/

HEADERS

Authorization token 80f4edf98e66878a25424f77fa245c1ecc146302

Example Request

create delete update get category

```
curl --location --request GET 'http://127.0.0.1:8000/api/classroom/category/' \
--header 'Authorization: token 80f4edf98e66878a25424f77fa245c1ecc146302'
```

3-module:

1-define in our serializer:

- a) which model that we need to serialise data from
- b) which fields that we need our serializer to handle

```
class
ModuleSerializer(serializers.ModelSerializer):
```

```
class Meta:  
    model=Module  
    fields='__all__'
```

2-inherit ModelViewSet class in our View Classes, and in each class we need to define in them

a) which serializer we need to use

```
class  
ModuleViewSet(viewsets.ModelViewSet):  
    queryset = Module.objects.all()  
    serializer_class = ModuleSerializer
```

4-page:

1-define in our serializer:

- a) which model that we need to serialise data from
- b) which fields that we need our serializer to handle

```

4 class PostFileContentSerializer(serializers.ModelSerializer):
5     class Meta:
6         model=PostFileContent
7         fields='__all__'
8 # get put update delete Page data
9 class PageSerializer(serializers.ModelSerializer):
10    class Meta:
11        model=Page
12        fields='__all__'
13 # get put update delete Comment data
14 class CommentSerializer(serializers.ModelSerializer):
15    class Meta:
16        model=Comment
17        fields='__all__'
18 # get put update delete Reply data
19 class ReplySerializer(serializers.ModelSerializer):
20    class Meta:
21        model=Reply
22        fields='__all__'

```

2-inherit ModelViewSet class in our View Classes, and in each class we need to define in them

a) which serializer we need to use

```

5 class PostFileContentViewSet(viewsets.ModelViewSet):
6     queryset = PostFileContent.objects.all()
7     serializer_class = PostFileContentSerializer
8
9 class PageViewSet(viewsets.ModelViewSet):
10    queryset = Page.objects.all()
11    serializer_class = PageSerializer
12
13 class CommentViewSet(viewsets.ModelViewSet):
14    queryset = Comment.objects.all()
15    serializer_class = CommentSerializer
16
17 class ReplyViewSet(viewsets.ModelViewSet):
18    queryset = Reply.objects.all()
19    serializer_class = ReplySerializer

```

5-assignment:

1-define in our serializer:

- a)which model that we need to serialize data from
- b)which fields that we need our serializer to handle

```

3 # get put update delete Assignment data
4 class AssignmentSerializer(serializers.ModelSerializer):
5     class Meta:
6         model=Assignment
7         fields='__all__'
8 # get put update delete Submission data
9 class SubmissionSerializer(serializers.ModelSerializer):
10    class Meta:
11        model=Submission
12        fields='__all__'
```

2-inherit ModelViewSet class in our View Classes, and in each class we need to define in them

- a)which serializer we need to use

```

5 class AssignmentViewSet(viewsets.ModelViewSet):
6     queryset = Assignment.objects.all()
7     serializer_class = AssignmentSerializer
8
9 class SubmissionViewSet(viewsets.ModelViewSet):
10    queryset = Submission.objects.all()
11    serializer_class = SubmissionSerializer
12
```

6-quiz:

1-define in our serializer:

- a)which model that we need to serialise data from
- b)which fields that we need our serializer to handle

```

4 class AnswerSerializer(serializers.ModelSerializer):
5     class Meta:
6         model=Answer
7         fields='__all__'
8 # get put update delete Question data
9 class QuestionSerializer(serializers.ModelSerializer):
10    class Meta:
11        model=Question
12        fields='__all__'
13 # get put update delete Quizzes data
14 class QuizzesSerializer(serializers.ModelSerializer):
15    class Meta:
16        model=Quizzes
17        fields='__all__'
18 # get put update delete Attempter data
19 class AttempterSerializer(serializers.ModelSerializer):
20    class Meta:
21        model=Attempter
22        fields='__all__'
```

2-inherit ModelViewSet class in our View Classes, and in each class we need to define in them

a) which serializer we need to use

```
5 class AnswerViewSet(viewsets.ModelViewSet):
6     queryset = Answer.objects.all()
7     serializer_class = AnswerSerializer
8
9 class QuestionViewSet(viewsets.ModelViewSet):
10    queryset = Question.objects.all()
11    serializer_class = QuestionSerializer
12
13 class QuizzesViewSet(viewsets.ModelViewSet):
14    queryset = Quizzes.objects.all()
15    serializer_class = QuizzesSerializer
16
17 class AttempterViewSet(viewsets.ModelViewSet):
18    queryset = Attempter.objects.all()
19    serializer_class = AttempterSerializer
```

7-completion:

1-define in our serializer:

a) which model that we need to serialise data from
b) which fields that we need our serializer to handle

```
4 class CompletionSerializer(serializers.ModelSerializer):
5     class Meta:
6         model=Completion
7         fields='__all__'
```

2-inherit ModelViewSet class in our View Classes, and in each class we need to define in them

a) which serializer we need to use

```
5 class CompletionViewSet(viewsets.ModelViewSet):
6     queryset = Completion.objects.all()
7     serializer_class = CompletionSerializer
```

8-schedule:

1-define in our serializer:

- a)which model that we need to serialise data from
- b)which fields that we need our serializer to handle

```
4 class AssignmentScheduleSerializer(serializers.ModelSerializer):  
5     class Meta:  
6         model=AssignmentSchedule  
7         fields='__all__'  
8 # get put update delete QuizSchedule data  
9 class QuizScheduleSerializer(serializers.ModelSerializer):  
10    class Meta:  
11        model=QuizSchedule  
12        fields='__all__'
```

2-inherit ModelViewSet class in our View Classes, and in each class we need to define in them

- a)which serializer we need to use

```
5 class AssignmentScheduleViewSet(viewsets.ModelViewSet):  
6     queryset = AssignmentSchedule.objects.all()  
7     serializer_class = AssignmentScheduleSerializer  
8  
9 class QuizScheduleViewSet(viewsets.ModelViewSet):  
10    queryset = QuizSchedule.objects.all()  
11    serializer_class = QuizScheduleSerializer
```

ISSUES & CHALLENGES

1-Learning python and the working pattern of django was very difficult. For those of us who are not familiar with django or python either was difficult in our first few steps.

2-Being new to Django (or frameworks in general), and the terms "projects" and "apps" and "api" and "ai integrate" were a little confusing at first.

3-With Django, a Project is just a collection of apps. Say you were building a zoom clone; you would create a project, "smarterdu zoom", that would contain apps like "live video", "comments", "share screen", etc.

4-Because Django doesn't serve media files itself, we had to separate the media files (images, JavaScripts, stylesheets, etc.) into a separate directory away from the apps.

5-Configuration of the local environment for Django is one of the biggest hitch. There are different versions of django available which require different specific versions of pip, virtualenv. For us as a beginner, it takes longer to install the framework locally than to build our first app.

Chapter 5

Mobile Application

Smart Edu App is done by a Mobile Framework called *Flutter* which helps to achieve the idea of the project with ease and produce the app for All mobile platforms.

Materials and Methods

Materials:

The main mobile app isn't a native mobile app so it's done by different stacks.

1. Dart Programming Language

A New Language introduced by Google in 2011, Dart is very popular to programmers as its syntax is similar to Java & C++ and you can use it to build mobile, desktop, server, and web applications. The Dart code compiles to native or JavaScript in order to run in the browser.

2. Flutter Framework

A Cutting Edge cross Platform Mobile Framework which has a lot of advantages in mobile development in generating with its software development kit (SDK). Flutter is an open source framework by Google for building beautiful, natively compiled, multi-platform applications from a single codebase.

Why did we choose Dart & Flutter?

One of the essential reasons why programmers love Flutter and prefer it over other frameworks is because of the programming language Dart. During the inception of Flutter, programmers analysed and evaluated many languages and ended up choosing Dart due to some surprising reasons and benefits it offers to the programming. Here are some of the reasons why Flutter uses Dart.

Flutter uses Dart as Dart allows Flutter to avoid the need for a separate declarative layout language like JSX and XML. The layout of Dart is declarative and programmatic, and it makes it easy for developers to read and visualise it very quickly and effortlessly. In addition to that, it makes it easy for Flutter to provide additional tooling as the layout is in one language and commonplace.

Another important reason why the duo is popular is that Dart, if required, uses the Just In Time compilation. This drastically reduces the time of development and responds faster.

What are the Advantages of Using Flutter rather than Native?

- **Faster in Development:** Flutter is a cross platform solution for mobile that means that developers do not have to write separate code to target each platform such as Android or iOS. Developers write one source base and Flutter compiles it into each Platform.
- **High Support from Google:** There are many competitors to Flutter around there. But Flutter has a big gap in Support and Community size. It is officially supported by *Google and Fast Growing Flutter Community*.

- **HHigh Performant:** Flutter introduces a new rendering engine which has proven an extraordinary speed in paint and render time. Load time and render time is not noticeable from Native Mobile apps.
- **No Depending on bridges or other pipelines:** other mobile cross platform frameworks use Bridge to compile the code to Native code.

The UI in Flutter:

The ui in flutter is a code with one language "dart" not like other Front-end frameworks like web(HTML, CSS, JAVASCRIPT) or mobile native(for example Android: XML with JAVA or KOTLIN). That make creating screens with flutter more easy.

Everything is a widget in flutter :

Widgets: Each element on a screen of the Flutter app is a widget. The view of the screen completely depends upon the choice and sequence of the widgets used to build the app. And the structure of the code of an app is a tree of widgets.

Category of Widgets:

There are mainly 14 categories in which the flutter widgets are divided. They are mainly segregated on the basis of the functionality they provide in a flutter application.

1. Accessibility: These are the set of widgets that make a flutter app more easily accessible.
2. Animation and Motion: These widgets add animation to other widgets.
3. Assets, Images, and Icons: These widgets take charge of assets such as display images and show icons.
4. Async: These provide async functionality in the flutter application.
5. Basics: These are the bundle of widgets that are absolutely necessary for the development of any flutter application.

6. Cupertino: These are the ios designed widgets.
7. Input: This set of widgets provides input functionality in a flutter application.
8. Interaction Models: These widgets are here to manage touch events and route users to different views in the application.
9. Layout: This bundle of widgets helps in placing the other widgets on the screen as needed.
10. Material Components: This is a set of widgets that mainly follow material design by Google.
11. Painting and effects: This is the set of widgets that apply visual changes to their child widgets without changing their layout or shape.
12. Scrolling: This provides scrollability of to a set of other widgets that are not scrollable by default.
13. Styling: This deals with the theme, responsiveness, and sizing of the app.
14. Text: This displays text.

Types of Widgets:

There are broadly two types of widgets in the flutter:

1. Stateless Widget
2. Stateful Widget

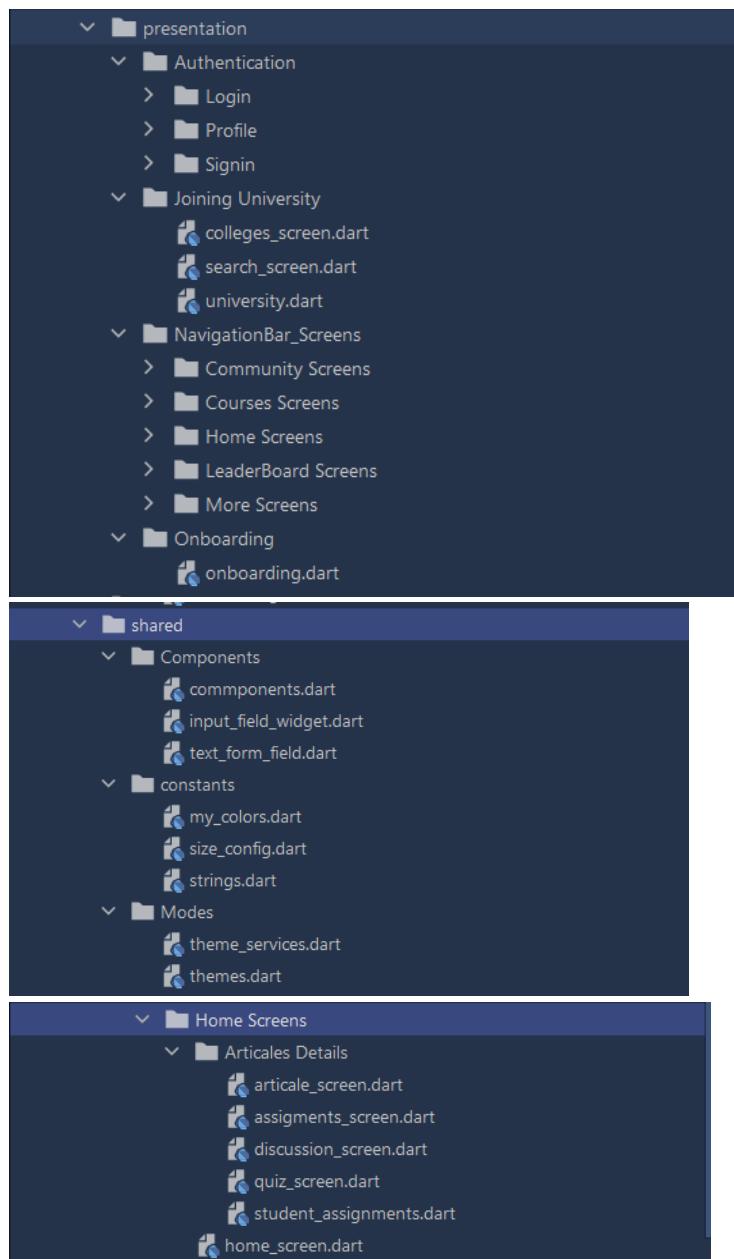
Screens:

Screen is the result of using some widgets with each other. It is builded with extends from state classes to handle all the screen states from the initial state and dispose state and all the other states.

Components:

Component is a reusable widget if we need to use a part of the ui from the design number of times we build a new file with this widget to call it up to any other view file.

Presentation layer structure:

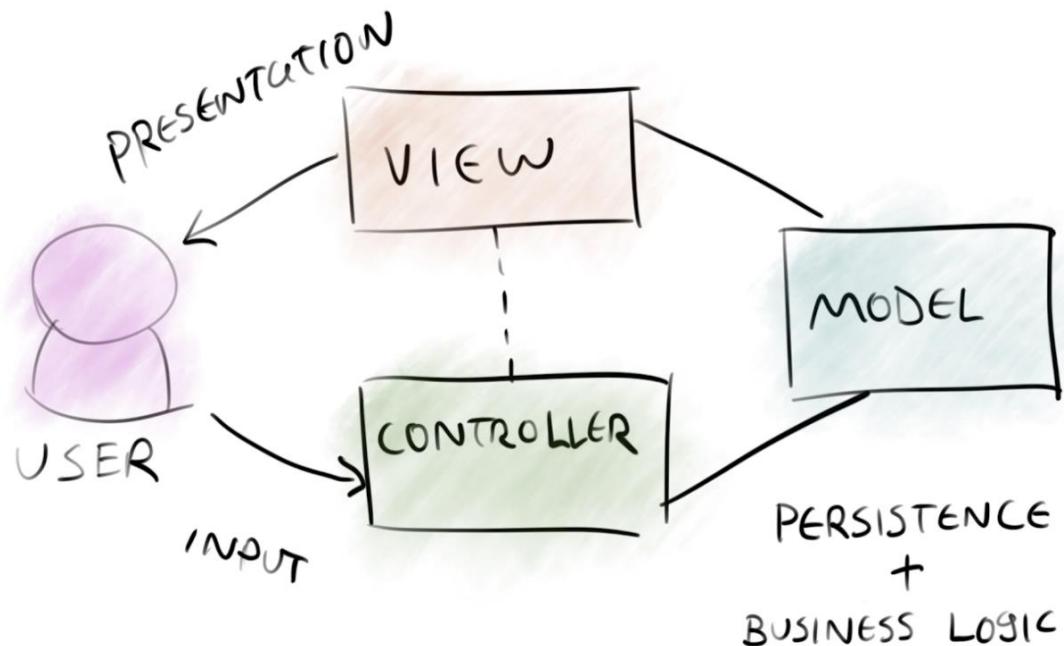




What are the Design Patterns & Code Styles we follow?

Flutter gives the developer the freedom to select the design pattern that fits the project business needs.

In the SmartEdu App we follow the Model View Controller Design pattern (MVC) . This pattern focuses on separating the business layer from the view layer from the model layer.



Model

The Model Layer stores data and its related logic. It represents data that is being transferred between controller components or any other related business logic. For example, a Controller object will retrieve the customer info from the database. It manipulates data and sends back to the database or uses it to render the same data. It responds to the request from the views and also responds to instructions from the controller to update itself. It is also the lowest level of the pattern which is responsible for maintaining data.

View

A View Layer is responsible for presenting the user interface.

Views commonly created by data layer when it is collected from the model data Layer. First the view will request the model to give data or information needed so that it presents the output presentation to the user.

The view Layer also represents the data from charts, diagrams, and tables. For example, any customer view will include all the UI components.

Controller

The Controller Layer is responsible for handling the user interaction. The controller interprets the user on Tap actions and callbacks and this will inform the model and the view to change as appropriate.

A Controller send's commands to the model to update its state for example (Login). The controller also sends commands to its associated view to change the view's presentation (For example Load Response Data from API).

• Tools

1. Code Editors & IDE's

The main software applications used to write dart code are *Android Studio* from Google or *Visual Studio Code* from Microsoft.

Android studio:

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA . On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices
- Apply Changes to push code and resource changes to your running app without restarting your app
- Code templates and GitHub integration to help you build common app features and import sample code
- Extensive testing tools and frameworks

- Lint tools to catch performance, usability, version compatibility, and other problems
 - C++ and NDK support
 - Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine
- User interface:
- The toolbar lets you carry out a wide range of actions, including running your app and launching Android tools.
 - The navigation bar helps you navigate through your project and open files for editing. It provides a more compact view of the structure visible in the Project window.
 - The editor window is where you create and modify code. Depending on the current file type, the editor can change. For example, when viewing a layout file, the editor displays the Layout Editor.
 - The tool window bar runs around the outside of the IDE window and contains the buttons that allow you to expand or collapse individual tool windows.
 - The tool windows give you access to specific tasks like project management, search, version control, and more. You can expand them and collapse them.
 - The status bar displays the status of your project and the IDE itself, as well as any warnings or messages.

2. Package Manager

The source of the code we need to add to the project as a bundled package. We use pub.io package manager.

3. Postman:

Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better APIs—faster.

Easily store, catalogue, and collaborate around all your API artefacts on one central platform. Postman can store and manage API specifications, documentation, workflow recipes, test cases and results, metrics, and everything else related to APIs.

The Postman platform includes a comprehensive set of tools that help accelerate the API lifecycle—from design, testing, documentation, and mocking to the sharing and discoverability of your APIs.

4. Simulators

In order to test the visuals of the app (UI) we need to run a visual device to see the changes in real time we use Android Emulators and iOS Simulators.

5. Packages

- bloc: is used for handling the state of the app and tracking any change of user data and user interaction.
- dio: is used for handling API and communicating with the network and database.
- path_provider: is used for giving us utilities to get the paths of devices.
- google_fonts: give us the fonts used in the app.
- shared_preferences: is used for saving user data and other required information needed in the app on the device.
- carousel_slider: is used for building the images beside each other and enabling swipe feature.
- video_player: is used for opening and running video players
- webview_flutter: is used for opening any link from the web in mobile.
- open_file: is used to open the PDF or video files from device storage or from link
- chewie: is a wrapper for video_player package to give nice functions and UI for app.

6. Firebase

Platform by Google that enables developers to develop mobile and Web apps databases. We use firebase for a realtime database for Chat Feature.

- **Methods and Procedures:**

- **Project Structure**

pubspec.yaml: This is the file we use to add metadata and configuration specific to our application. With this file's help, we can configure dependencies such as image assets, fonts, and app versions.

Content Example:

```
name: smartEdu
description: A Graduation Project for the Zagazig University 2022
```

Lib: This is the most important folder in the project, used to write most of the dart code. By default, the lib folder contains the main.dart file, which is the application's entry point. This configuration, however, can be changed.

pubspec.lock: This file contains the version of each dependency and packages used in the flutter application.

Content Example:

```
packages:
  archive:
    dependency: transitive
    description:
      name: archive
      url: "https://pub.dartlang.org"
    source: hosted
    version: "3.1.11"
```

main.dart: This is the main entry of the app

Content Example:

```
void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  DioHelper.init();
  await CacheHelper.init();
  bool? isDark = CacheHelper.getData('isDark');
  bool? isRtl = CacheHelper.getData('isRtl');
  bool? showOnBoard = CacheHelper.getData('ShowOnBoard');
```

```
runApp(MyApp(isDark: isDark, isRtl: isRtl, startWidget: Home())));
}
```

Flutter State Management

In this section, we are going to discuss [state management](#) and how we can [handle](#) it in the Flutter. We know that in Flutter, everything is a [widget](#). The widget can be classified into [two categories](#),

- Stateless widget
- Stateful widget.

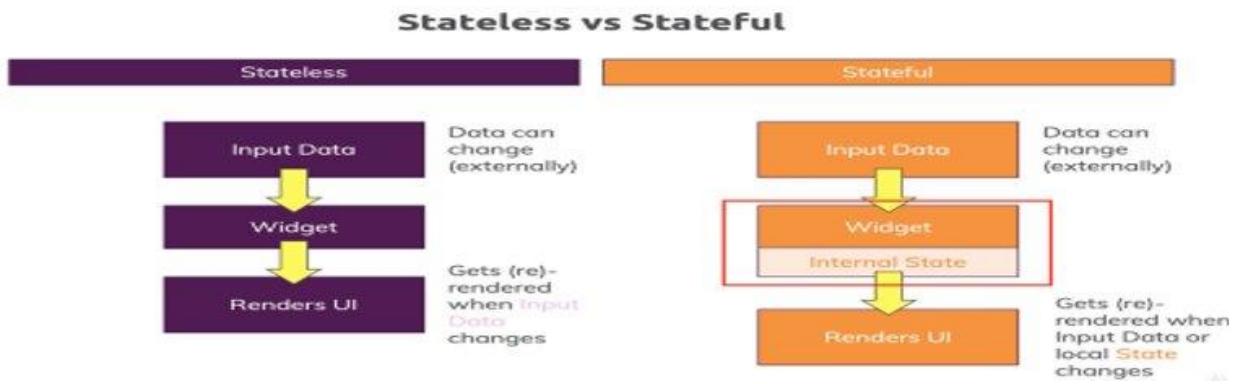
What is State?

A state is information that can be read when the widget is built and might change or modified over a lifetime of the app. If you want to change your widget, you need to update the state object, which can be done by using the `setState()` function available for Stateful widgets. The `setState()` function allows us to set the properties of the state object that triggers a redraw of the UI.

Stateless widget Vs Stateful widget.

Stateless widget does not have any internal state. It means once it is built, we cannot change or modify it until they are initialized again.

Stateful widget is dynamic and has a state. It means we can modify it easily throughout its lifecycle without reinitialized it again.



Build UI from application state

The **state management** is one of the most popular and necessary processes in the lifecycle of an application. According to official documentation, Flutter is declarative. It means Flutter builds its UI by **reflecting** the current state of your app. The following figure explains it more clearly where you can build a **UI** from the application state..



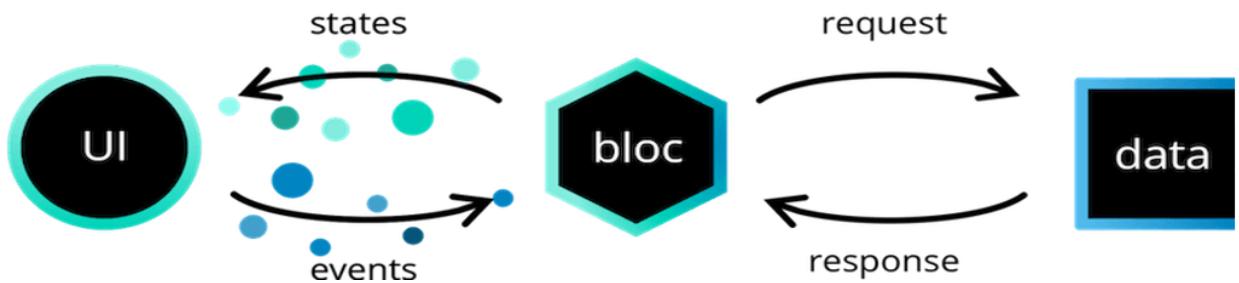
What is Bloc?

Bloc is a design pattern created by Google to help separate business logic from the presentation layer and enable a developer to reuse code more efficiently.

A state management library called Bloc was created and maintained by Felix Angelo. It helps developers implement the Bloc design pattern in their Flutter application. It means that a developer must know the state of an app at any time. There should be something displayed on the screen for every interaction with the app to let users know what is happening.

Technically, for every interaction inside the application, there should be a state emerging from it. For example, when the data is fetching, the app should be in a

loading state displaying a loading animation on the screen. When the internet is off, the application should display a pop-up to let the user know there is no internet connection.



BASIC CONCEPTS

Events: Events are input to a bloc, they're usually added as a result of user activities like button pushes, or lifecycle events like page loads. You can model your event as anything, from a primitive data type, such as an integer, to any complex abstracted classes.

```
abstract class AuthEvent {}
class LoginEvent extends AuthEvent {}
class LogoutEvent extends AuthEvent {}
```

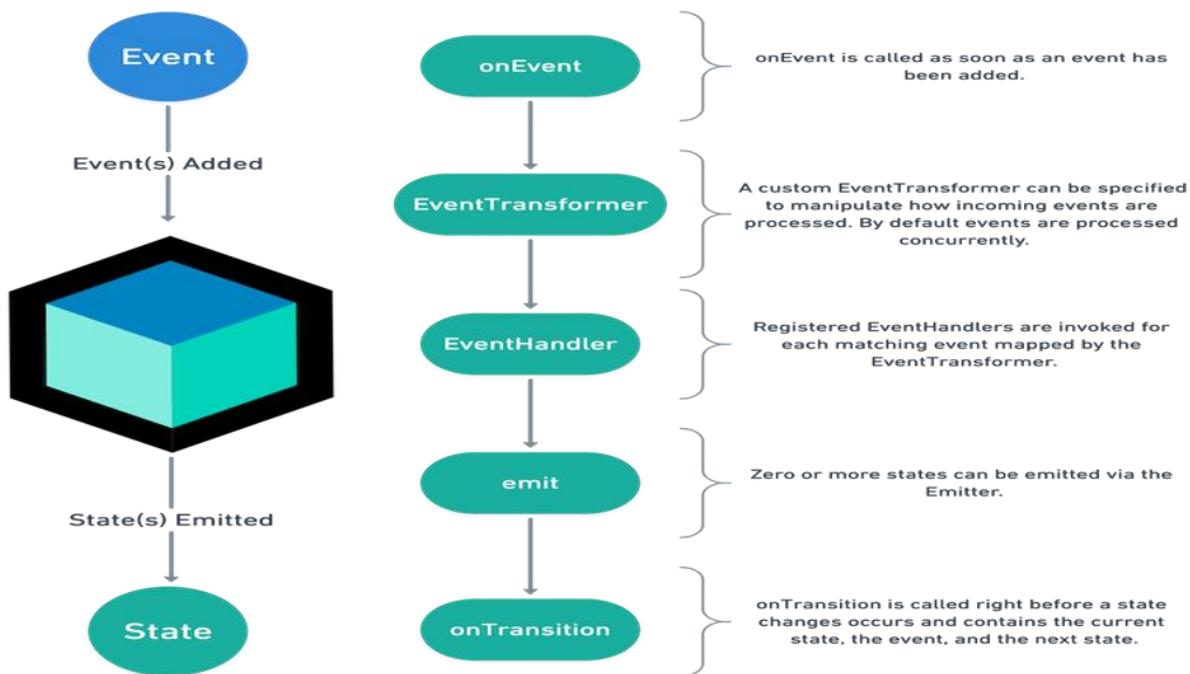
States: States are an output of a bloc, they represent the application state. The UI components listen to a state change and redraw a portion of themselves based on the current state. The state can also be modelled as anything from a primitive data type, such as an integer, to any complex abstracted classes.

```
abstract class AuthState {}
class UnAuthenticatedState extends AuthState {}
class AuthenticatedState extends AuthState {}
```

Bloc Flow

State changes in bloc begin when events are added which triggers `onEvent`. The events are then funnelled through an `EventTransformer`. By default, each event is processed concurrently but a custom `EventTransformer` can be provided to manipulate the incoming event stream. All registered `EventHandlers` for that event type are then invoked with the incoming event. Each `EventHandler` is responsible for emitting zero or more states in response to the event. Lastly, `onTransition` is

called just before the state is updated and contains the current state, event, and next state.



Adding Bloc to Flutter

First, you need to add the following dependency to the **pubspec.yaml** file:

```
dependencies:  
  flutter_bloc: ^8.0.1
```

Import it

```
import 'package:flutter_bloc/flutter_bloc.dart';
```

The bloc dependency is used because we also used cubit in the example. The flutter_bloc will provide you with the widgets necessary to make it easier to use the Bloc pattern in Flutter.

BLOC WIDGETS

The bloc widgets help to rebuild/notify the UI components in response to a state change. Since cubit is just a subclass of bloc, you can use cubit anywhere that a bloc is required when working with bloc widgets.

BLOC PROVIDER

BlocProvider is a Flutter widget that uses BlocProvider.of <T> to supply a bloc to its children. It's a dependency injection (DI) widget that lets you deliver a single instance of a bloc to many widgets within a subtree. BlocProvider should be used to create new blocs that will be available to the remainder of the subtree in most cases. Because BlocProvider is in charge of generating the bloc in this situation, it will also be in charge of closing it.

You will almost always want to put the BlocProvider above the MaterialApp so that it will be available everywhere in your application.

```
BlocProvider(  
    create: (BuildContext context) => AuthBloc(),  
    child: MaterialApp(  
        title: 'Flutter Bloc Demo',  
        theme: ThemeData(  
            primarySwatch: Colors.blue,  
            visualDensity: VisualDensity.adaptivePlatformDensity,  
        ),  
        home: MyHomePage(),  
    ),  
) ;
```

With this, you can use BlocProvider.of <AuthBloc>(context) to get the instance of the auth bloc, and add an event like this;

```
BlocProvider.of<AuthBloc>(context).add(LoginEvent());
```

Multi BLOC PROVIDER

MultiBlocProvider is a Flutter widget that combines the functionality of numerous BlocProvider widgets into a single widget. MultiBlocProvider enhances readability by removing the requirement for numerous BlocProviders to be nested.

You can have all of these BlocProviders inside a single MultiBlocProvider like this

```
MultiBlocProvider(  
    providers: [
```

```

    BlocProvider(
      create: (BuildContext context) => FirstBloc(),
    ) ,
    BlocProvider(
      create: (BuildContext context) => SecondBloc(),
    ) ,
    BlocProvider(
      create: (BuildContext context) => ThirdBloc(),
    ) ],
  child: ChildWidget(),
),
),

```

Observing Bloc Changes

The BlocOverrides class contains the property blocObserver which will enable us to observe any change in the Bloc and this would make it easier when getting stuck on some issue. Now create a file called todo_bloc_observer.dart and add the following:

```

import 'package:bloc/bloc.dart';

class TodoBlocObserver extends BlocObserver {
  @override
  void onEvent(Bloc bloc, Object? event) {
    super.onEvent(bloc, event);
    print('onEvent $event');
  }

  @override
  void onChange(BlocBase bloc, Change change) {
    super.onChange(bloc, change);
    print('onChange $change');
  }

  @override
  void onTransition(Bloc bloc, Transition transition) {
    super.onTransition(bloc, transition);
    print('onTransition $transition');
  }

  @override
  void onError(BlocBase bloc, Object error, StackTrace stackTrace) {
    print('onError $error');
    super.onError(bloc, error, stackTrace);
  }
}

```

- **API**

What is API?

API stands for Application Programming interface which is considered a software intermediary that allows two applications to talk to each other. Each time you use an app like Facebook, send an instant message, or check the weather on your phone, you're using an API.

Any application on our mobile phone, connects to the Internet and sends data to a server. The server then retrieves that data, interprets it, performs the necessary actions and sends it back to your phone. The application then interprets that data and presents you with the information you wanted in a readable way. This is what an API is - all of this happens via API. In our situation we connect to the server to get the data about available courses and student assignments, ...etc.

We talk about data in many sections which means that *data* is everything around us. We handle different types of data in apps and from multiple sources.

In App we preview data from static data or from a file and from an API which we will talk about deeply.

How does the API work?

A User can make GET, POST, PUT, or DELETE HTTP (CRUD) requests to a database. In return, the database sends us data, results, or responses in the form of JSON, HTML, or XML (with the JSON format being the most widely used). We then parse the JSON into a proper model class and use it in our app.

Modern APIs adhere to standards (HTTP and REST), that are developer-friendly, easily accessible and understood broadly

How do we Integrate API into our App?

To communicate with servers using http protocols we could use native http requests in dart but we instead choose a package called **dio** that creates a layer on top of native http and introduces easy to use methods to create CRUD calls.

First: create class to make integration & Initialization with api using Dio package.

```
class DioHelper {  
    static Dio dio;  
    static init() {  
        dio = Dio(  
            BaseOptions(  
                baseUrl: SERVER_URL,  
                receiveDataWhenStatusError: true,  
                connectTimeout: 20 * 1000,  
                receiveTimeout: 20 * 1000,  
            ),  
        );  
  
    }  
}
```

Second: Add CRUD methods in the class

1. GET

```
static Future<Response> get({  
    required String url,  
    Map<String, dynamic>? query,  
    String? token,  
    Map<String, dynamic>? data,  
) async {  
    dio.options.headers = {  
        'Content-Type': 'application/json',  
        'Authorization': 'Token $token'  
    };  
}
```

```
        return await dio.get(url,queryParameters: query,);
    }
```

2. POST

```
static Future<Response> post({
    required String url,
    Map<String, dynamic>? query,
    Map<String, dynamic>? data,
    String? token,
}) async {
    dio.options.headers = {
        'Content-Type': 'application/json',
        'Authorization': LOGIN==url ? null : 'Token $token'
    };
    return await dio.post(url, queryParameters: query, data: data,);
}
```

3. PUT

```
static Future<Response> put(
    {required String url,
    Map<String, dynamic>? query,
    Map<String, dynamic>? data,
    String? token}) async {
    dio.options.headers = {
        'Content-Type': 'application/json',
        'Authorization': 'Token ${admin_token}'
    };
    return await dio.put(url, queryParameters: query, data: data,);
}
```

4. DELETE

```
static Future<Response> delete({
    required String url,
    String lang = 'en',
    String? token,
```

```

}) async {
  dio.options.headers = {
    'lang': lang,
    'Content-Type': 'application/json',
    'Authorization': 'Token $token'
  };
  return await dio.delete(url);
}
}

```

API endpoints

Endpoint is the url that we send to the CRUD methods as url to save it
 we create a file to store endpoints which use it as a url to get or post data

```

Class EndPoints {
  static const ADMINLOGIN = '/login/';
  static const TeacherREGISTER = '/signup/teacher/';
  static const LOGOUT = 'logout/';
  static const LOGIN = '/login/';
  static const StudentREGISTER = '/signup/student/';
  static const courses = "/courses/";
  static const courseDetails = "/course/";
  static const chapters = "/chapters/";
  static const chapterDetails = "/chapter/";
}

```

Caching

we create the storage file to store user data using SharedPreferences

We just create a method which contains all methods to abbreviation our time

First method which save data it helps us to store our data of any data type(int, string, bool ,double)

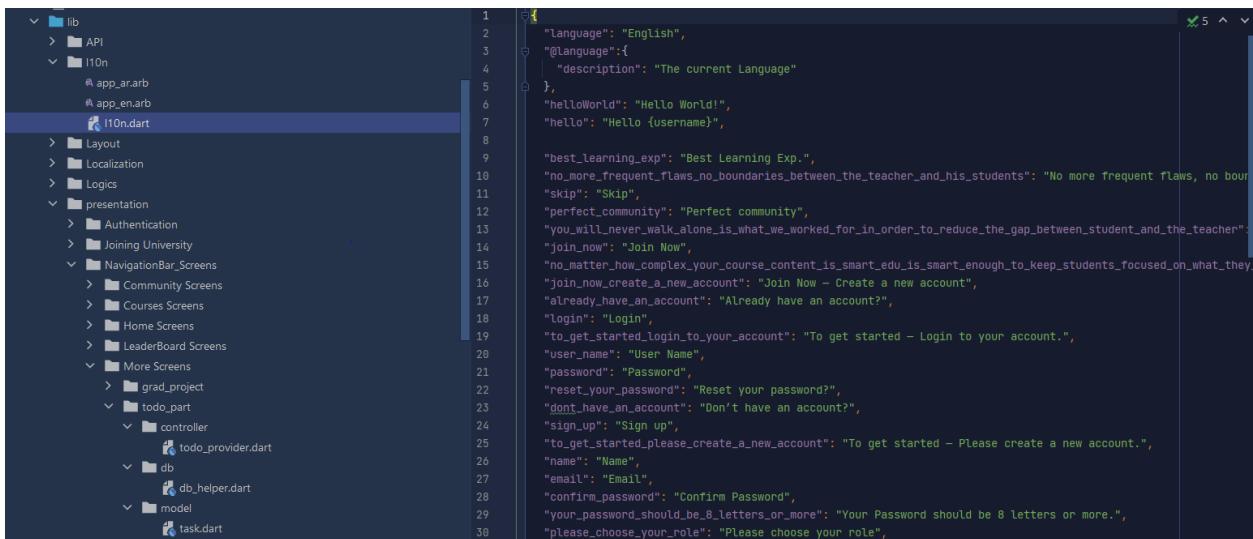
Second method which get data it helps us to get the data which we stored locally we just call it

Third method which is remove data it helps us to remove data from device

```
class CacheHelper {  
    static SharedPreferences? sharedPreferences;  
    static init() async {  
        sharedPreferences = await SharedPreferences.getInstance();  
    }  
    static dynamic getData(String key) {  
        return sharedPreferences!.get(key);  
    }  
  
    static Future<bool> saveData(  
        {required String key, required dynamic value}) async {  
        if (value is String) return await sharedPreferences!.setString(key, value);  
        if (value is int) return await sharedPreferences!.setInt(key, value);  
        if (value is bool) return await sharedPreferences!.setBool(key, value);  
        return await sharedPreferences!.setDouble(key, value);  
    }  
  
    static Future<bool> removeData(String key) async {  
        return await sharedPreferences!.remove(key);  
    }  
}
```

Localization:

We create 3 files to make translation in our app (L10n >> ar , en)



The image shows a file explorer on the left and a code editor on the right. The file explorer displays a project structure with folders like lib, API, and I10n containing files such as app_ar.arb, app_en.arb, and I10n.dart. The code editor shows an English localization file (app_en.arb) with the following content:

```
1 "language": "English",
2 "@Language": {
3   "description": "The current Language"
4 },
5 "helloWorld": "Hello World!",
6 "hello": "Hello {username}",
7
8 "best_learning_exp": "Best Learning Exp.",
9 "no_more_frequent_flaws_no_boundaries_between_the_teacher_and_his_students": "No more frequent flaws, no boundaries between the teacher and his students",
10 "skip": "Skip",
11 "perfect_community": "Perfect community",
12 "you_will_never_walk_alone_is_what_we_worked_for_in_order_to_reduce_the_gap_between_student_and_the_teacher": "You will never walk alone is what we worked for in order to reduce the gap between student and the teacher",
13 "join_now": "Join Now",
14 "no_matter_how_complex_your_course_content_is_smart_edu_is_smart_enough_to_keep_students_focused_on_what_they": "No matter how complex your course content is smart.edu is smart enough to keep students focused on what they",
15 "join_now_create_a_new_account": "Join Now - Create a new account",
16 "already_have_an_account": "Already have an account?",
17 "login": "Login",
18 "to_get_started_login_to_your_account": "To get started - Login to your account",
19 "user_name": "User Name",
20 "password": "Password",
21 "reset_your_password": "Reset your password?",
22 "don_t_have_an_account": "Don't have an account?",
23 "sign_up": "Sign up",
24 "to_get_started_please_create_a_new_account": "To get started - Please create a new account.",
25 "name": "Name",
26 "email": "Email",
27 "confirm_password": "Confirm Password",
28 "your_password_should_be_8_letters_or_more": "Your Password should be 8 letters or more.",
29 "please_choose_your_role": "Please choose your role",
30
```

Ar file this helps us to store arabic translation

EN file this helps us to store english translation

We just call the word which we need like this

```
AppLocalizations.of(context)!.join_now_create_a_new_account,
```

and it auto make the translation which depend on app language

Generating a numerical systems quizzes:

- No repeated questions
- No repeated from system to another
- Changeable number of quizzes models

- Changeable number of questions in the quiz
- For int and double numbers

The image displays two side-by-side screenshots of a mobile application interface. Both screens show a blue header bar with the text "System Converting".

Left Screen (Top):

- Time: 11:09
- Signal strength: 4 bars
- Network: LTE2
- Battery: 100%

Left Screen (Bottom):

Conversion problem:

$$\begin{array}{r} 7 \\ \hline 4 & 50 \\ \hline 3 \end{array}$$

Options:

- 1) 39
- 2) 11
- 3) 44
- 4) 16
- 5) 47

Answer: 1) 39

Right Screen (Top):

- Time: 11:09
- Signal strength: 4 bars
- Network: LTE2
- Battery: 100%

Right Screen (Bottom):

Conversion problem:

Q: How to convert (10010) from Binary to Octal ?

- 1) 22
- 2) 60
- 3) 54
- 4) 17
- 5) 45

Answer: 1) 22

Q: How to convert (18) from Decimal to Octal ?

- 1) 22
- 2) 35
- 3) 40
- 4) 12
- 5) 47

Answer: 1) 22

Q: How to convert (8) from Decimal to Binary ?

- 1) 111
- 2) 1000

Navigation icons at the bottom of both screens include vertical ellipsis, square, and left arrow.

Chat Rooms

1. Initialise Firebase

```
await Firebase.initializeApp(
  options: DefaultFirebaseOptions.currentPlatform,
);
```

2. Define Models and References

```
// get the rooms references to the database
DatabaseReference roomRef = FirebaseDatabase.instance.ref('rooms');
```

```
// define Model to the room chat and User
class Message {
    final User sender;
    final String time;
    final String text;
    final bool isRead;

    // The message of chat data
    Message({
        required this.sender,
        required this.time,
        required this.text,
        required this.isRead,
    });
}

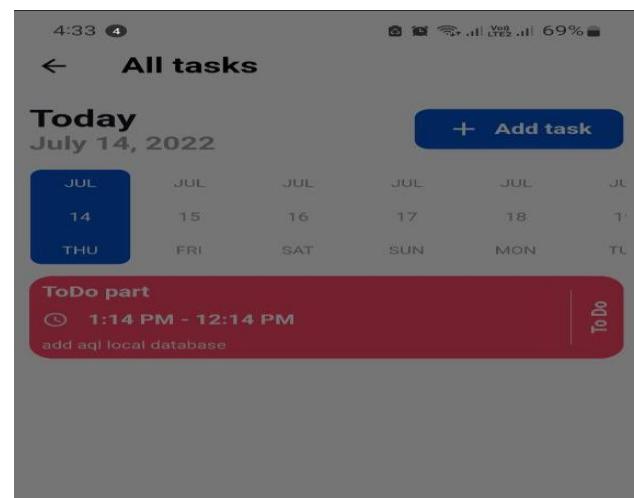
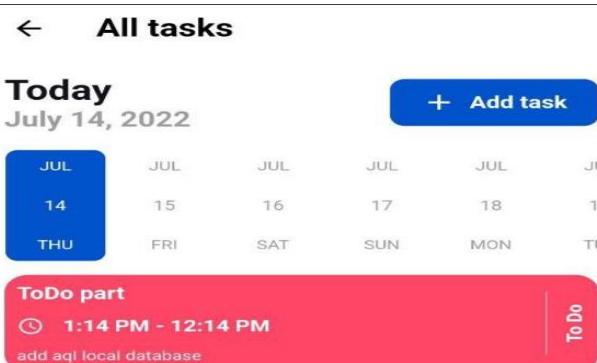
// The User info saved in the chat
class User {
    final int id;
    final String name;

    User({
        required this.id,
        required this.name,
    });
}
```

To-do Tasks:

To stay organized is sometimes the hardest thing to do like a lot of things are going on in our lives. You can't keep track of it all and most of us can't either which is why there are to-do lists. Be it our professional or personal life, there are things that we usually forget, and then comes the need for such an app that can keep track of it all.

The Interface:



The page where old and current tasks are displayed, as well as adding new tasks. It is also possible to modify the pre-existing tasks, whether after completing it or deleting it.

Add new task UI :

Here a new task is added with its data in terms of the date and time specified to complete this task

The screenshot shows a mobile application interface for adding a new task. At the top, there is a back arrow labeled "←" and the text "add task". Below this is a "Title" field containing "ToDo part". Under "Title", there is a "Note" section with the text "add sql local database". The next section is "Date", showing "Thursday, July 14, 2022" with a calendar icon. Below "Date" are "Start date" and "End date" fields, both set to "1:14 PM" with clock icons. Under "Repeat", it says "None" with a dropdown arrow. In the "Colors" section, there are five colored circles: blue, pink (with a checkmark), yellow, and green. To the right of the colors is a red button with a white plus sign and the text "Create task". At the bottom of the screen, there is a navigation bar with three icons: three horizontal lines, a circle, and a left arrow.

Create a new DB for a new task:

The screenshot shows a code editor with a sidebar containing project files and a main pane displaying Dart code. The code defines a `DbHelper` class with methods for initializing the database and inserting tasks.

```
1 import ...
3
4 class DbHelper {
5     static Database? _db;
6     static const int _version = 1;
7     static const String _tableName = 'tasks';
8
9     static Future<void> initDb() async {
10        if (_db != null) {
11            return;
12        }
13        try {
14            String _path = await getDatabasesPath() + 'tasks.db';
15            _db =
16                await openDatabase(_path, version: _version, onCreate: (db, version) {
17                    print('creating a new table');
18                    db.execute(
19                        'CREATE TABLE $_tableName('
20                            'id INTEGER PRIMARY KEY AUTOINCREMENT, '
21                            'title STRING, date STRING, startTime STRING, endTime STRING, repeat STRING, '
22                            'note TEXT, remind INTEGER, color INTEGER, isCompleted INTEGER)');
23                });
24        } catch (e) {
25            print('this is db error: $e');
26        }
27    }
28
29
30
31    static Future<int> insertTask(Task task) async{
32        print('inserte done');
33    }
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
59
60
61
62
63
64
65
66
67
68
69
69
70
71
72
73
74
75
76
77
78
79
79
80
81
82
83
84
85
86
87
88
89
89
90
91
92
93
94
95
96
97
97
98
98
99
99
100
100
101
101
102
102
103
103
104
104
105
105
106
106
107
107
108
108
109
109
110
110
111
111
112
112
113
113
114
114
115
115
116
116
117
117
118
118
119
119
120
120
121
121
122
122
123
123
124
124
125
125
126
126
127
127
128
128
129
129
130
130
131
131
132
132
133
133
134
134
135
135
136
136
137
137
138
138
139
139
140
140
141
141
142
142
143
143
144
144
145
145
146
146
147
147
148
148
149
149
150
150
151
151
152
152
153
153
154
154
155
155
156
156
157
157
158
158
159
159
160
160
161
161
162
162
163
163
164
164
165
165
166
166
167
167
168
168
169
169
170
170
171
171
172
172
173
173
174
174
175
175
176
176
177
177
178
178
179
179
180
180
181
181
182
182
183
183
184
184
185
185
186
186
187
187
188
188
189
189
190
190
191
191
192
192
193
193
194
194
195
195
196
196
197
197
198
198
199
199
200
200
201
201
202
202
203
203
204
204
205
205
206
206
207
207
208
208
209
209
210
210
211
211
212
212
213
213
214
214
215
215
216
216
217
217
218
218
219
219
220
220
221
221
222
222
223
223
224
224
225
225
226
226
227
227
228
228
229
229
230
230
231
231
232
232
233
233
234
234
235
235
236
236
237
237
238
238
239
239
240
240
241
241
242
242
243
243
244
244
245
245
246
246
247
247
248
248
249
249
250
250
251
251
252
252
253
253
254
254
255
255
256
256
257
257
258
258
259
259
260
260
261
261
262
262
263
263
264
264
265
265
266
266
267
267
268
268
269
269
270
270
271
271
272
272
273
273
274
274
275
275
276
276
277
277
278
278
279
279
280
280
281
281
282
282
283
283
284
284
285
285
286
286
287
287
288
288
289
289
290
290
291
291
292
292
293
293
294
294
295
295
296
296
297
297
298
298
299
299
300
300
301
301
302
302
303
303
304
304
305
305
306
306
307
307
308
308
309
309
310
310
311
311
312
312
313
313
314
314
315
315
316
316
317
317
318
318
319
319
320
320
321
321
322
322
323
323
324
324
325
325
326
326
327
327
328
328
329
329
330
330
331
331
332
332
333
333
334
334
335
335
336
336
337
337
338
338
339
339
340
340
341
341
342
342
343
343
344
344
345
345
346
346
347
347
348
348
349
349
350
350
351
351
352
352
353
353
354
354
355
355
356
356
357
357
358
358
359
359
360
360
361
361
362
362
363
363
364
364
365
365
366
366
367
367
368
368
369
369
370
370
371
371
372
372
373
373
374
374
375
375
376
376
377
377
378
378
379
379
380
380
381
381
382
382
383
383
384
384
385
385
386
386
387
387
388
388
389
389
390
390
391
391
392
392
393
393
394
394
395
395
396
396
397
397
398
398
399
399
400
400
401
401
402
402
403
403
404
404
405
405
406
406
407
407
408
408
409
409
410
410
411
411
412
412
413
413
414
414
415
415
416
416
417
417
418
418
419
419
420
420
421
421
422
422
423
423
424
424
425
425
426
426
427
427
428
428
429
429
430
430
431
431
432
432
433
433
434
434
435
435
436
436
437
437
438
438
439
439
440
440
441
441
442
442
443
443
444
444
445
445
446
446
447
447
448
448
449
449
450
450
451
451
452
452
453
453
454
454
455
455
456
456
457
457
458
458
459
459
460
460
461
461
462
462
463
463
464
464
465
465
466
466
467
467
468
468
469
469
470
470
471
471
472
472
473
473
474
474
475
475
476
476
477
477
478
478
479
479
480
480
481
481
482
482
483
483
484
484
485
485
486
486
487
487
488
488
489
489
490
490
491
491
492
492
493
493
494
494
495
495
496
496
497
497
498
498
499
499
500
500
501
501
502
502
503
503
504
504
505
505
506
506
507
507
508
508
509
509
510
510
511
511
512
512
513
513
514
514
515
515
516
516
517
517
518
518
519
519
520
520
521
521
522
522
523
523
524
524
525
525
526
526
527
527
528
528
529
529
530
530
531
531
532
532
533
533
534
534
535
535
536
536
537
537
538
538
539
539
540
540
541
541
542
542
543
543
544
544
545
545
546
546
547
547
548
548
549
549
550
550
551
551
552
552
553
553
554
554
555
555
556
556
557
557
558
558
559
559
560
560
561
561
562
562
563
563
564
564
565
565
566
566
567
567
568
568
569
569
570
570
571
571
572
572
573
573
574
574
575
575
576
576
577
577
578
578
579
579
580
580
581
581
582
582
583
583
584
584
585
585
586
586
587
587
588
588
589
589
590
590
591
591
592
592
593
593
594
594
595
595
596
596
597
597
598
598
599
599
600
600
601
601
602
602
603
603
604
604
605
605
606
606
607
607
608
608
609
609
610
610
611
611
612
612
613
613
614
614
615
615
616
616
617
617
618
618
619
619
620
620
621
621
622
622
623
623
624
624
625
625
626
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
```

Video Meeting

Meet & Chat



New Meeting



Join Meeting



Join a Meeting

Room ID

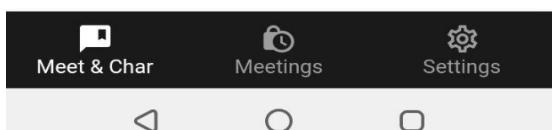
Abdelhamed Nouh

Join

Mute Audio

Turn Off My Video

Create/Join Meetings with just a click!





Creating Meeting

```
void createMeeting({
    required String roomName,
    required bool isAudioMuted,
    required bool isVideoMuted,
    String username = '',
}) async {
    try {
        FeatureFlag featureFlag = FeatureFlag();
        featureFlag.welcomePageEnabled = false;
        featureFlag.resolution = FeatureFlagVideoResolution
            .MD_RESOLUTION; // Limit video resolution to 360p
        String name;
        if (username.isEmpty) {
            name = _authMethods.user.displayName!;
        } else {
            name = username;
        }
        var options = JitsiMeetingOptions(room: roomName)
            ..userDisplayName = name
            ..userEmail = _authMethods.user.email
            ..userAvatarURL = _authMethods.user.photoURL
            ..audioMuted = isAudioMuted
            ..videoMuted = isVideoMuted;

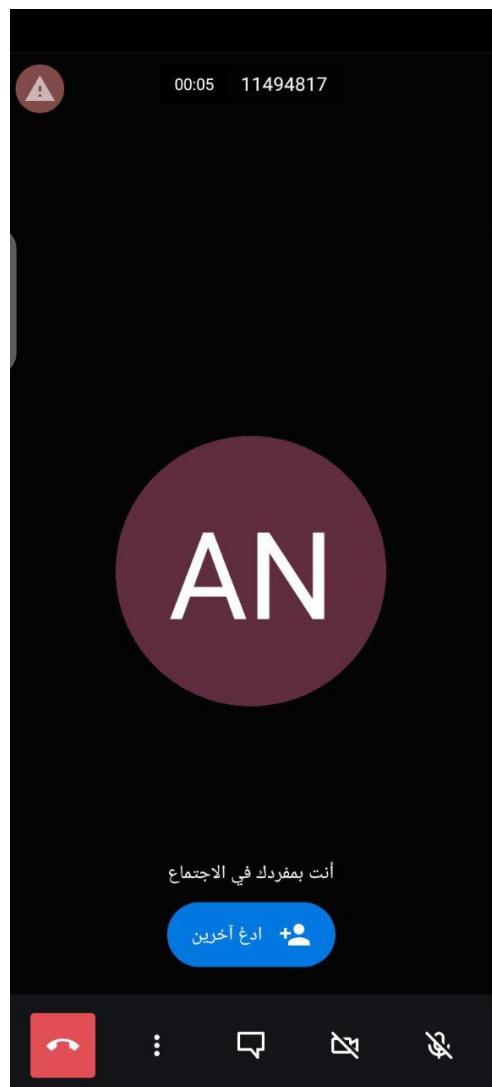
        _firestoreMethods.addToMeetingHistory(roomName);
        await JitsiMeet.joinMeeting(options);
    } catch (error) {
        print("error: $error");
    }
}
```

Activate Window
Go to Settings

New meeting code:

```
1 import 'dart:math';
2
3 import 'package:flutter/material.dart';
4 import '../resources/jitsi_meet_methods.dart';
5 import '../widgets/home_meeting_button.dart';
6
7 class MeetingScreen extends StatelessWidget {
8   MeetingScreen({Key? key}) : super(key: key);
9
10   final JitsiMeetMethods _jitsiMeetMethods = JitsiMeetMethods();
11
12   createNewMeeting() async {
13     var random = Random();
14     String roomName = (random.nextInt(10000000) + 10000000).toString();
15     _jitsiMeetMethods.createMeeting(
16       roomName: roomName, isAudioMuted: true, isVideoMuted: true);
17   }
18
19   joinMeeting(BuildContext context) {
20     Navigator.pushNamed(context, '/video-call');
21   }
22
23   @override
24   Widget build(BuildContext context) {
25     return Column(
26       children: [
27         Row(
28           mainAxisAlignment: MainAxisAlignment.spaceEvenly,
29           children: [
30             HomeMeetingButton(
31               onPressed: createNewMeeting,
32               text: 'New Meeting',
33             ),
34           ],
35         ),
36       ],
37     );
38   }
39 }
40
41 
```

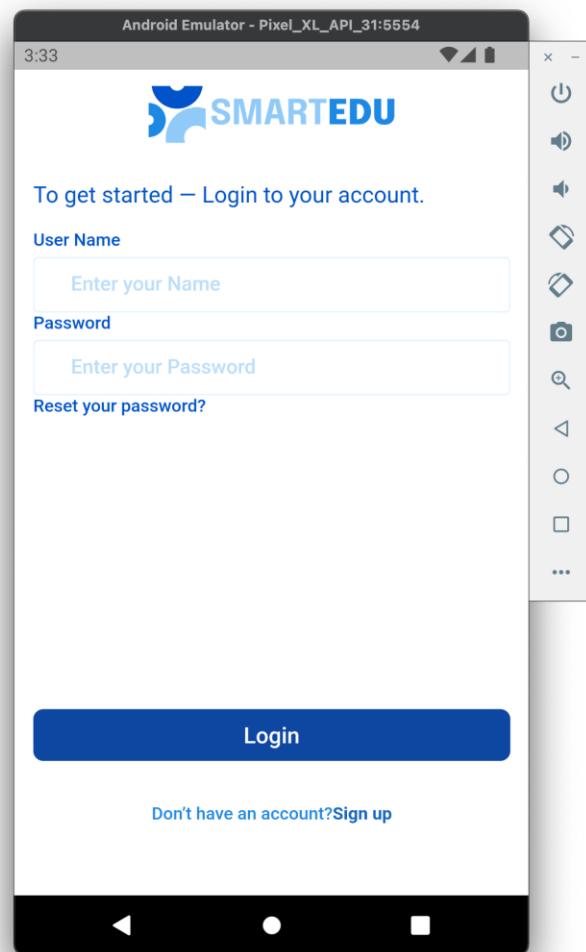
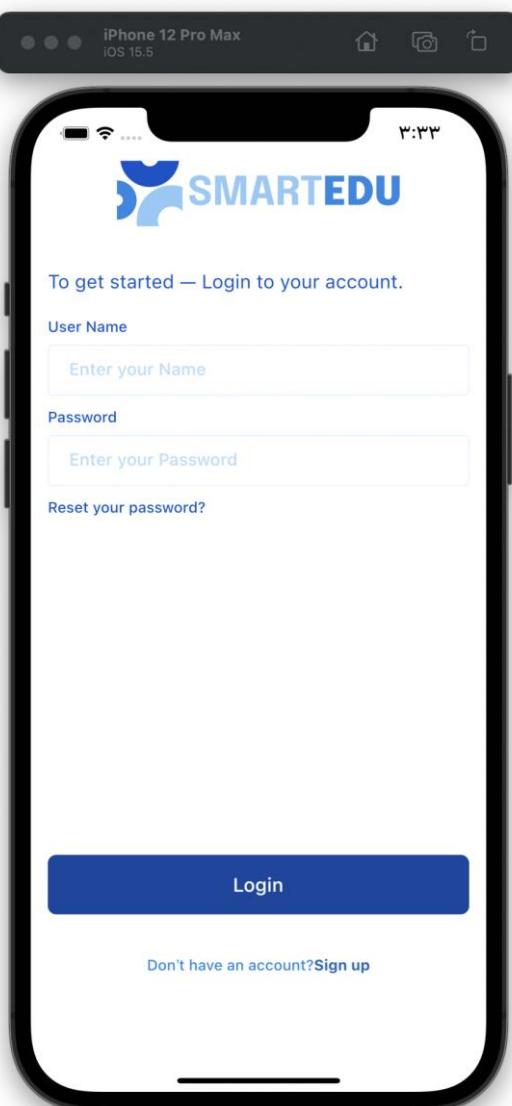
Activate Windows
Go to Settings to activate Windows.



This is what the meeting page looks like

Results and Discussion

Cross Platform Mobile App



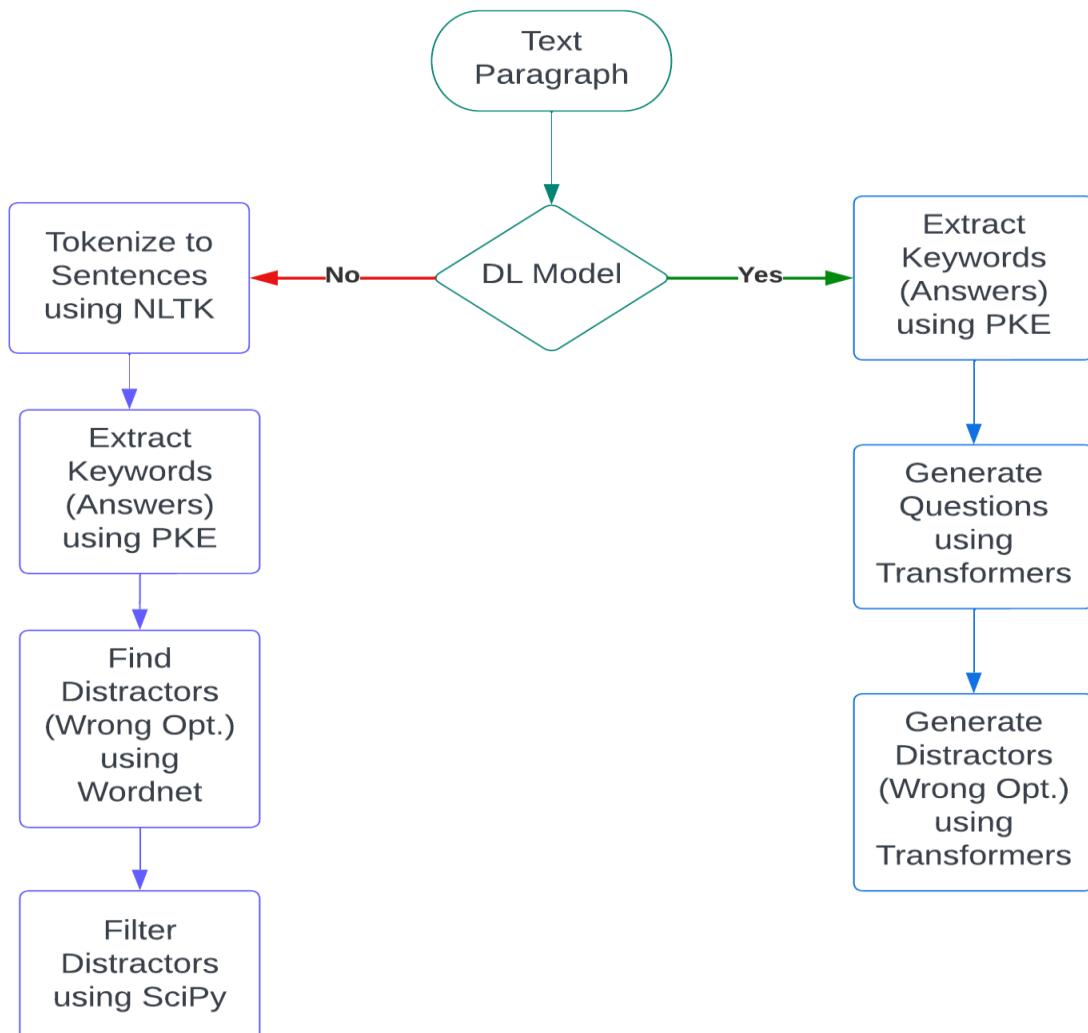
Chapter 6

Artificial Intelligence

Quiz Generator

During our project, We went through Different ways to generate MCQ and True/False questions from a text. In the first term, we used different NLP techniques to generate both MCQ and T/F questions we called it **NLP Model**. In the second term, we used the power of transformers and combine it with deep learning to be suitable for our tasks we called it **DL Model**.

MCQ Quiz



DL Model:

Transformers Theory

NLP

Natural Language Processing is a field of linguistics and machine learning focused on understanding everything related to human language. The aim of NLP tasks is not only to understand single words individually but to be able to understand the context of those words.

Why is it challenging?

Computers don't process information in the same way as humans. For example, when we read the sentence "I am hungry," we can easily understand its meaning. Similarly, given two sentences such as "I am hungry" and "I am sad," we're able to easily determine how similar they are. For machine learning (ML) models, such tasks are more difficult. The text needs to be processed to enable the model to learn from it. And because language is complex, we need to think carefully about how this processing must be done.

Transformer models are used to solve all kinds of NLP tasks,

The following is a list of common NLP tasks, with some examples of each:

Classifying whole sentences: Getting the sentiment of a review, detecting if an email is spam, determining if a sentence is grammatically correct or whether two sentences are logically related or not

Classifying each word in a sentence: Identifying the grammatical components of a sentence (noun, verb, adjective), or the named entities (person, location, organization)

Generating text content: Completing a prompt with auto-generated text, filling in the blanks in a text with masked words

Extracting an answer from a text: Given a question and a context, extracting the answer to the question based on the information provided in the context

Generating a new sentence from an input text: Translating a text into another language, summarizing a text

This list is far from comprehensive and is just meant to highlight a few of the different kinds of Transformer models. Broadly, they can be grouped into three categories:

GPT-like (also called auto-regressive Transformer models)

BERT-like (also called auto-encoding Transformer models)

BART/T5-like (also called sequence-to-sequence Transformer models)

We will dive into these families in more depth later on.

Transformers are language models

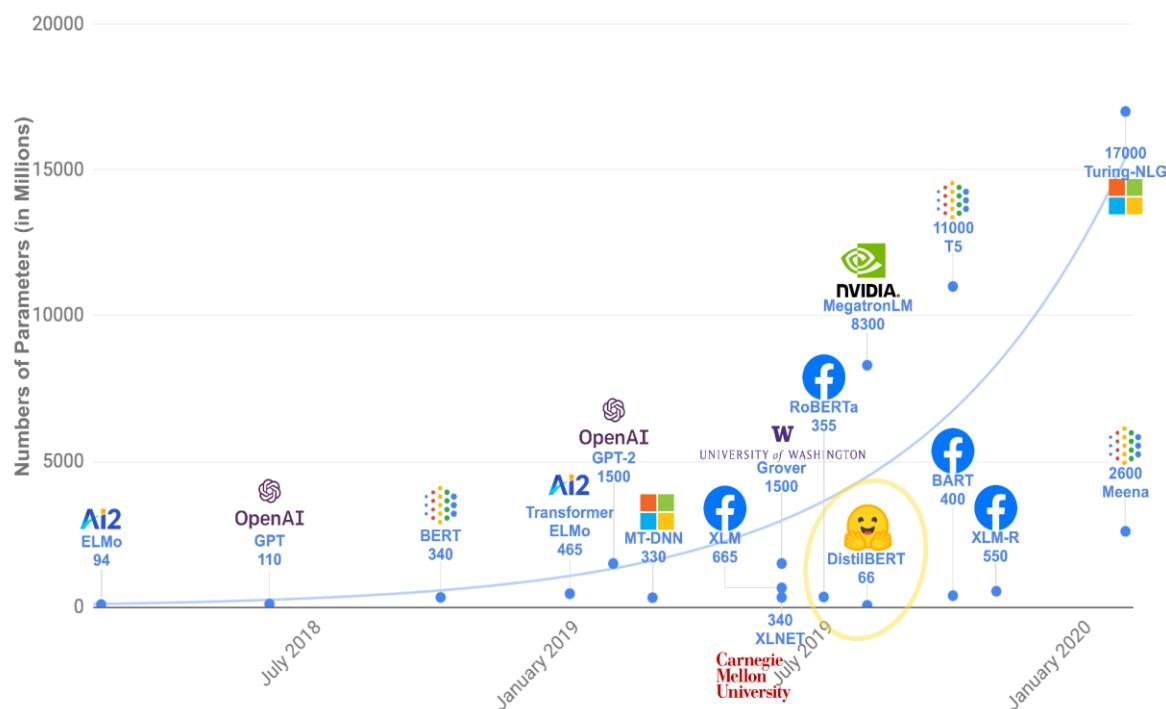
All the Transformer models mentioned above (GPT, BERT, BART, T5, etc.) have been trained as language models. This means they have been trained on large amounts of raw text in a self-supervised fashion. Self-supervised learning is a type of

training in which the objective is automatically computed from the model's inputs. That means that humans are not needed to label the data!

This type of model develops a statistical understanding of the language it has been trained on, but it's not very useful for specific practical tasks. Because of this, the general pre-trained model then goes through a process called transfer learning. During this process, the model is fine-tuned in a supervised way — that is, using human-annotated labels — on a given task.

Transformers are big models

Apart from a few outliers (like DistilBERT), the general strategy to achieve better performance is by increasing the models' sizes as well as the amount of data they are pre-trained on.

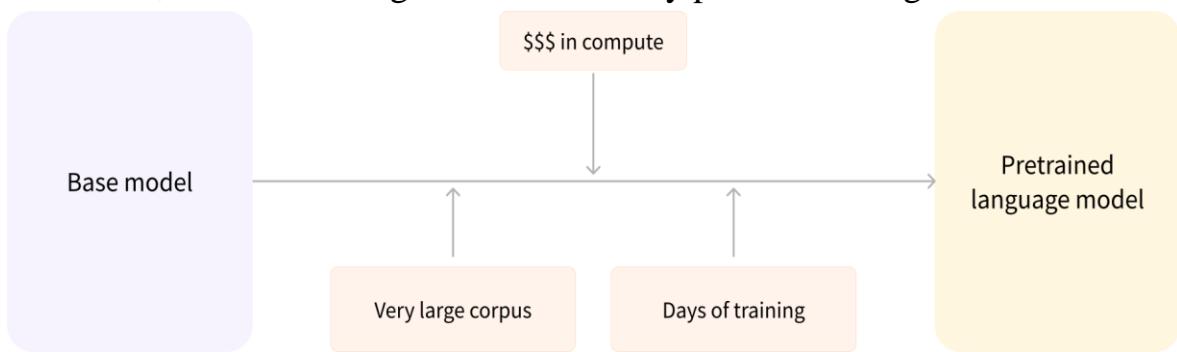


Unfortunately, training a model, especially a large one, requires a large amount of data. This becomes very costly in terms of time and compute resources. It even translates to environmental impact.

This is why sharing language models is paramount: sharing the trained weights and building on top of already trained weights reduces the overall compute cost and carbon footprint of the community.

Transfer Learning

Pretraining is the act of training a model from scratch: the weights are randomly initialized, and the training starts without any prior knowledge.

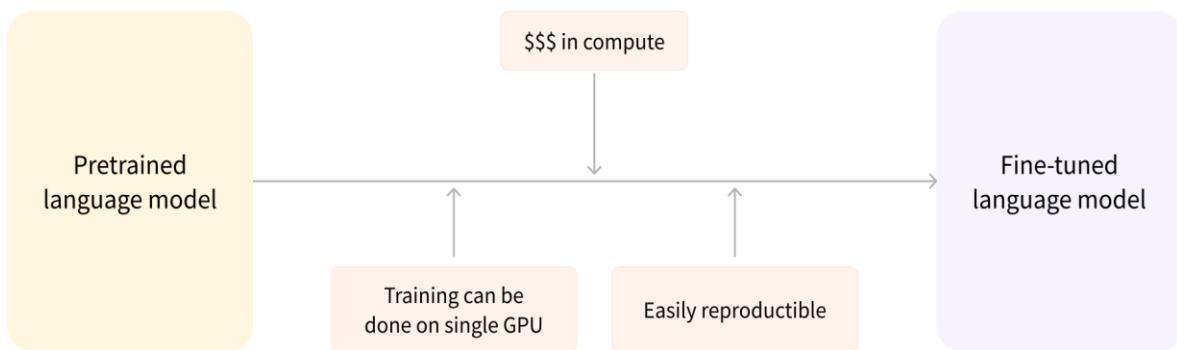


This pretraining is usually done on very large amounts of data. Therefore, it requires a very large corpus of data, and training can take up to several weeks.

Fine-tuning, on the other hand, is the training done after a model has been pretrained. To perform fine-tuning, you first acquire a pretrained language model, then perform additional training with a dataset-specific to your task. Wait — why not simply train directly for the final task? There are a couple of reasons:

- The pretrained model was already trained on a dataset that has some similarities with the fine-tuning dataset. The fine-tuning process is thus able to take advantage of the knowledge acquired by the initial model during pretraining (for instance, with NLP problems, the pretrained model will have some kind of statistical understanding of the language you are using for your task).
- Since the pretrained model was already trained on lots of data, the fine-tuning requires way less data to get decent results.

For the same reason, the amount of time and resources needed to get good results are much lower.



Fine-tuning a model, therefore, has lower time, data, financial, and environmental costs. It is also quicker and easier to iterate over different fine-tuning schemes, as the training is less constraining than a full pretraining.

This process will also achieve better results than training from scratch (unless you have lots of data), which is why you should always try to leverage a pretrained model — one as close as possible to the task you have at hand — and fine-tune it.

Architecture

The model is primarily composed of two blocks:

Encoder (left): The encoder receives input and builds a representation of it (its features). This means that the model is optimized to acquire understanding from the input.

Decoder (right): The decoder uses the encoder's representation (features) along with other inputs to generate a target sequence. This means that the model is optimized for generating outputs.

Each of these parts can be used independently, depending on the task:

- **Encoder-only models:** Good for tasks that require understanding of the input, such as sentence classification and named entity recognition.
- **Decoder-only models:** Good for generative tasks such as text generation.
- **Encoder-decoder models or sequence-to-sequence models:** Good for generative tasks that require input, such as translation or summarization.

Attention layer

this layer will tell the model to pay specific attention to certain words in the sentence you passed it (and more or less ignore the others) when dealing with the representation of each word.

The attention mask can also be used in the encoder/decoder to prevent the model from paying attention to some special words — for instance, the special padding word used to make all the inputs the same length when batching together sentences.

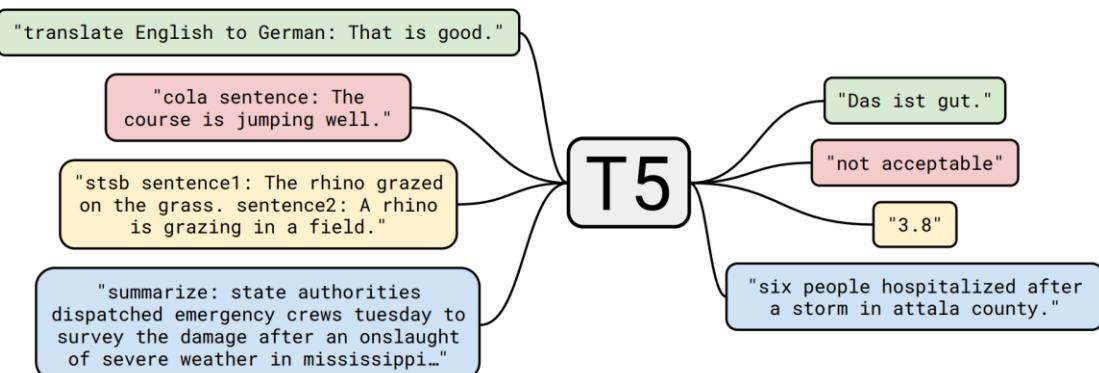
Models Training

We used T5 transformer as a pretrained model then we trained it on new datasets to get our models ready for the desired tasks, Questions Generation and Distractors Generation

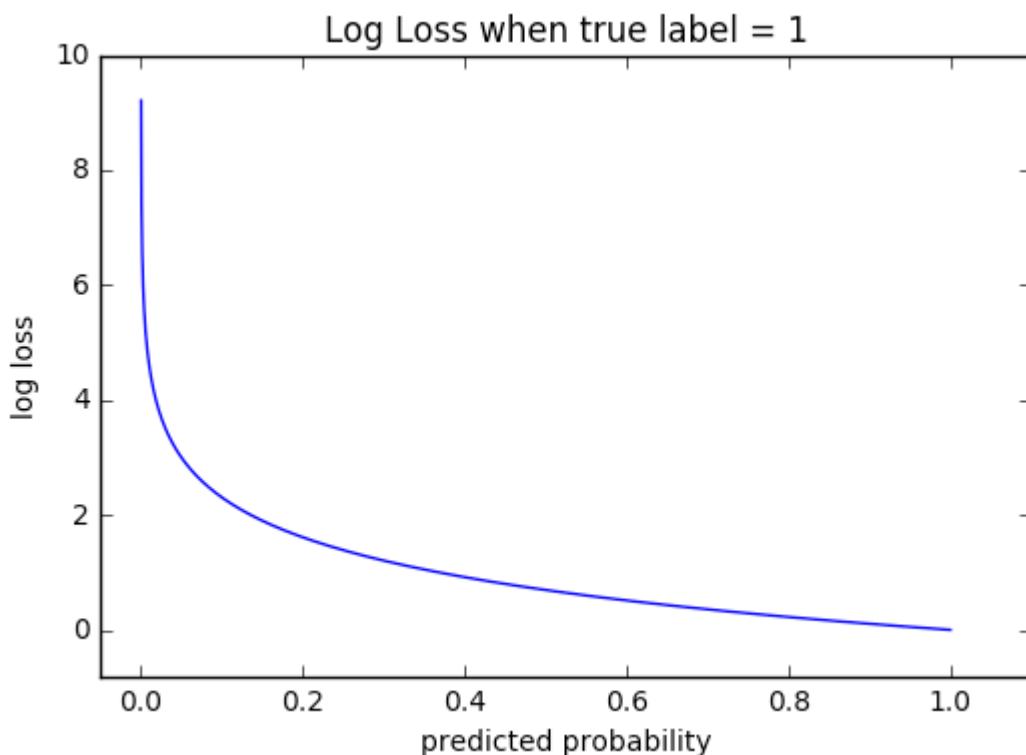


T5 Transformer

T5 is an encoder-decoder model pre-trained on a multi-task mixture of unsupervised and supervised tasks and for which each task converts all NLP problems into a text-to-text format. It is trained using teacher forcing. This means that for training, we always need an input sequence and a corresponding target sequence.



T5 uses Cross Entropy Loss to compute the loss of the model



SQuAD (Stanford Question Answering Dataset)

SQuAD is a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage, or the question might be unanswerable. We used SQuAD1.1 which contains 100,000 questions with their answers and context.

RACE Dataset

Race is a large-scale reading comprehension dataset with more than 28,000 passages and nearly 100,000 questions. The dataset is collected from English examinations in China, which are designed for middle school and high school students. The dataset can be served as the training and test sets for machine comprehension.

SciQ Dataset

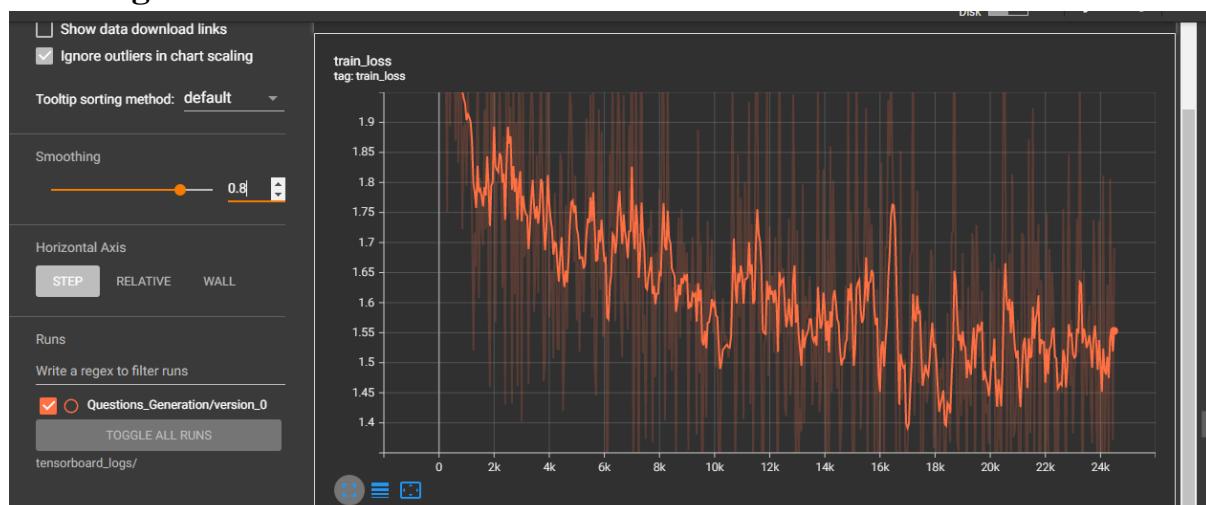
The SciQ dataset contains 13,679 crowdsourced science exam questions about Physics, Chemistry and Biology, among others. The questions are in multiple-choice format with 4 answer options each. For the majority of the questions, an additional paragraph with supporting evidence for the correct answer is provided.

Training Results

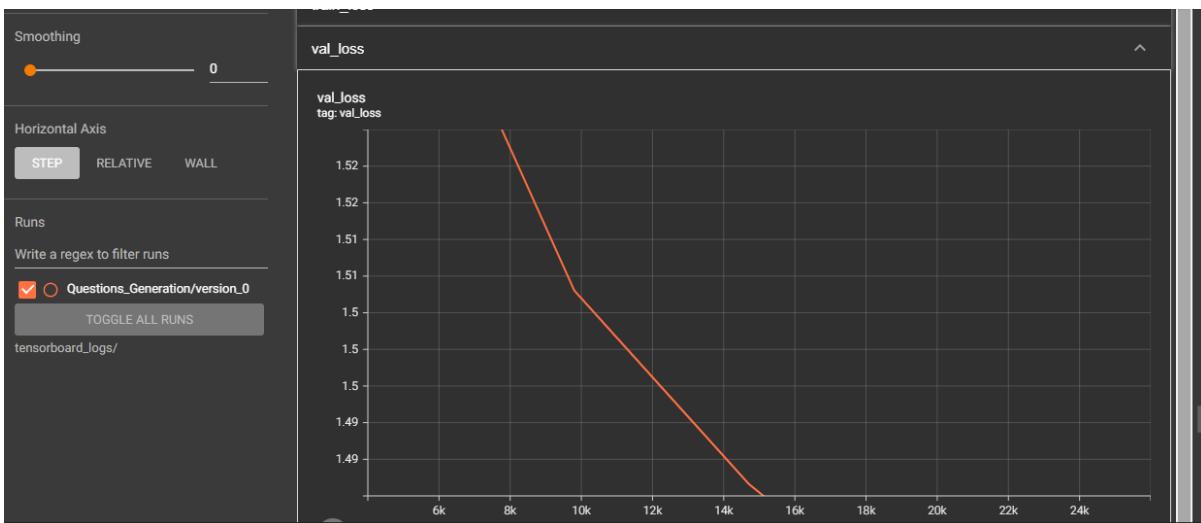
Question Generation

We trained data with 16 batches, 5 epochs, and $5 * 10^{-5}$ learning rate

Training loss:



Validation Loss:

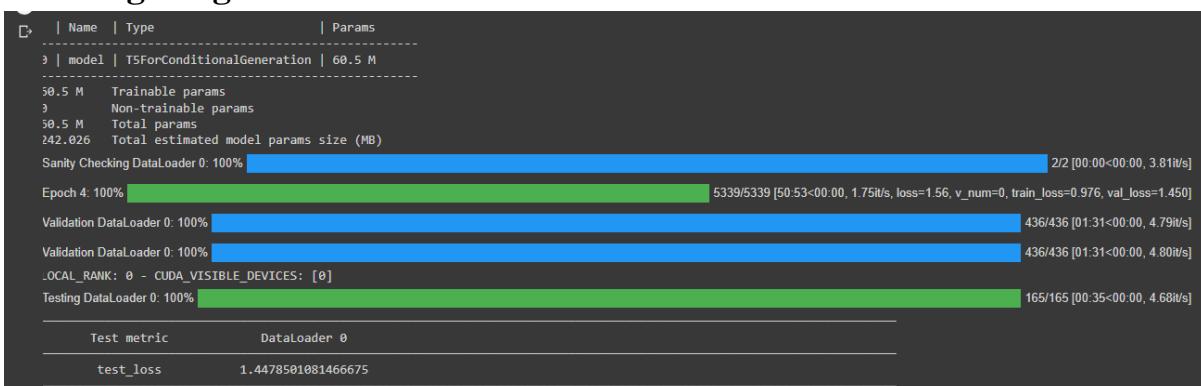


Test Overall Loss:

Loss = 1.448



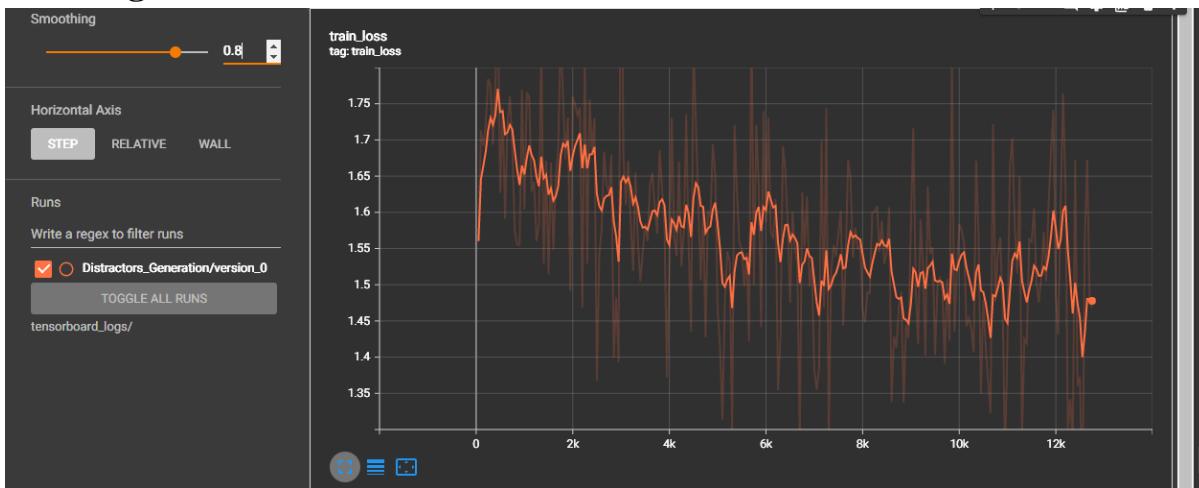
Training Progress Bar



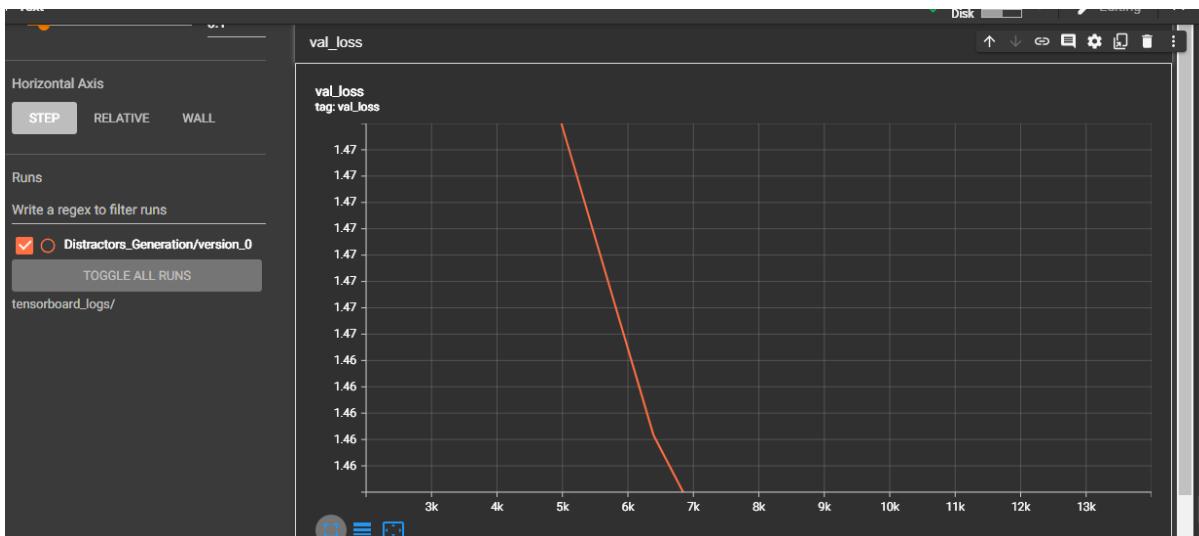
Distractors Generation

We combine two datasets Race and Sciq then trained the total dataset with 24 batches, 9 epochs, and $1 * 10^{-4}$ learning rate

Training loss:



Validation Loss

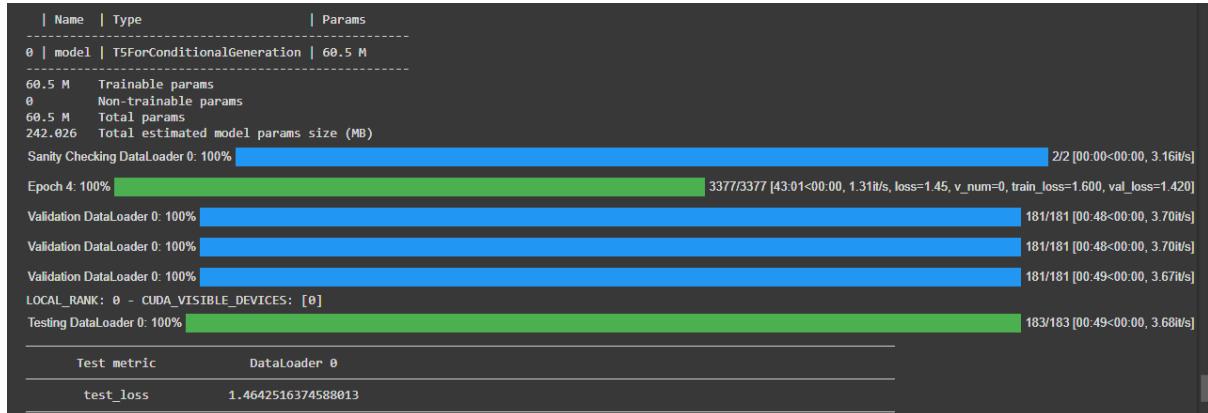


Loss Overall Loss

Loss = 1.464



Training Progress Bar



Models Output

Ex:

Context: It is a common misconception that the Sun is yellow, or orange or even red. However, the Sun is essentially all colors mixed together, which appear to our eyes as white. Rainbows are light from the Sun, separated into its colors. Each color in the rainbow (red, orange, yellow, green, blue, violet) has a different wavelength. Red is the longest, blue the shortest.

Answer: Yellow

Question Generation

```
for beam_output in beam_outputs:  
    sent = Qtokenizer.decode(beam_output, skip_special_tokens=True,clean_up_tokenization_spaces=True)  
    print(sent)  
  
output = sent  
  
question: What is the common misconception that the Sun is?  
question: What is a common misconception that the Sun is?  
question: What color is a common misconception that the Sun is?
```

Distractors Generation

```
for beam_output1 in beam_outputs1:  
    sent1 = Dtokenizer.decode(beam_output1, skip_special_tokens=True,clean_up_tokenization_spaces=True)  
    print(sent1)  
  
distractor1: red distractor2: green distractor3: green  
distractor1: red distractor2: green distractor3: red  
distractor1: red distractor2: green distractor3: blue  
distractor1: green distractor2: red distractor3: green  
distractor1: red distractor2: orange distractor3: green
```

NLP Model:

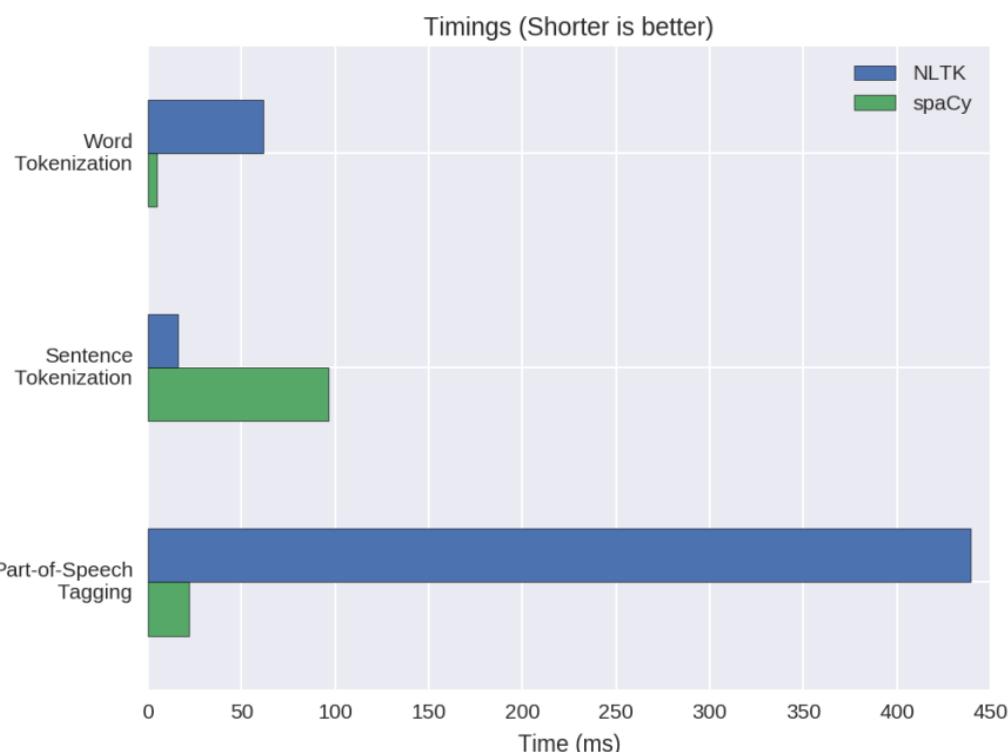
It receives text paragraphs as input and splits this text into some sentences then it gets a keyword in each sentence to be the answer and removes it from the sentence by replacing it with blank space. The model takes the keyword to find its co-hyponyms and finally, it computes the distance between the keyword and its co-hyponyms to choose the best distractors.

Tokenize to Sentence using NLTK:

NLTK: Natural Language Toolkit

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning.

NLTK vs SpaCy



As we can see above, in word tokenization and POS-tagging spaCy performs better, but in sentence tokenization, NLTK outperforms spaCy. Its poor performance in sentence tokenization is a result of differing approaches: NLTK attempts to split the text into sentences. In contrast, spaCy constructs a syntactic tree for each sentence, a more robust method that yields much more information about the text.

Implementation

```
# natural language toolkit, provides lexical resources and text processing libraries
import nltk
nltk.download('punkt') # used in tokenization
# tokenizers divide strings into lists of substrings
from nltk.tokenize import sent_tokenize
# use tokenizer to segment text and return list of sentences
def sent_tokenizer(text):
    sentences=[]
    for s in sent_tokenize(text) :
        sentences.append(s)
    return sentences
```

Results

Ex1: It is a common misconception that the Sun is yellow, or orange or even red. However, the Sun is essentially all colors mixed together, which appear to our eyes as white. Rainbows are light from the Sun, separated into its colors. Each color in the rainbow (red, orange, yellow, green, blue, violet) has a different wavelength. Red is the longest, blue the shortest.

Ex2: programming is a collaboration between humans and computers. programming is giving a set of instructions to a computer to execute. java is one of the most popular programming language in the world. java is easy to learn and simple to use.java is open-source and free. java is secure, fast and powerful. java has a huge community support. Java is an object oriented language which gives a clear structure to programs and allows code to be reused, lowering development costs. Computer systems have an indirect impact on third parties. computers can cause loss of information and resources that might result severely harmful for users. The data used to fit the model is called the training data

Ex1:

```
[ 'It is a common misconception that the Sun is yellow, or orange or even red.',  
  'However, the Sun is essentially all colors mixed together, which appear to our eyes as white.',  
  'Rainbows are light from the Sun, separated into its colors.',  
  'Each color in the rainbow (red, orange, yellow, green, blue, violet) has a different wavelength.',  
  'Red is the longest, blue the shortest.' ]
```

Ex2:

```
[ 'programming is a collaboration between humans and computers.',  
  'programming is giving a set of instructions to a computer to execute.',  
  'java is one of the most popular programming language in the world.',  
  'java is easy to learn and simple to use.java is open-source and free.',  
  'java is secure, fast and powerful.',  
  'java has a huge community support.',  
  'Java is an object oriented language which gives a clear structure to programs and allows code to be reused, lowering development costs.',  
  'Computer systems have an indirect impact on third parties.',  
  'computers can cause loss of information and resources that might result severely harmful for users.',  
  'The data used to fit the model is called the training data']
```

Extract Keywords (Answers) using PKE:

PKE

PKE is an open-source python-based keyphrase extraction toolkit. The pke module provides an end-to-end keyphrase extraction pipeline in which each component can be easily modified or extended to develop new approaches.

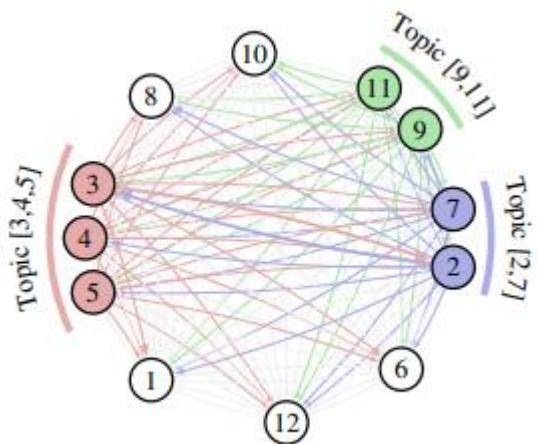
Theory

We used unsupervised keyphrase extraction with Multipartite Graphs

Unsupervised means there is no labeled data. No prior training. It can work on any new document.

A complete directed multipartite graph is built, in which nodes are keyphrase candidates that are connected only if they belong to different topics. weight edges according to the distance between two candidates.

These nodes (keywords) are ranked by exploiting co-occurrence relationships
and the top 10 candidates are returned as ranked candidates.



Implementation

```
import pke      # keyphrase extraction toolkit
import string  # define string constants like punctuation
def get_keyword(text):
    out_list=[]
    try:
        # create a MultipartiteRank extractor.
        extractor = pke.unsupervised.MultipartiteRank()
        stoplist = list(string.punctuation)
        stoplist += pke.lang.stopwords.get('en')
        # load the content of the document.
        extractor.load_document(input=text, stoplist=stoplist)
        # select the longest sequences of nouns and adjectives, that do
        # not contain punctuation marks or stopwords as candidates.
        pos = {'NOUN', 'PROPN', 'ADJ'}
        extractor.candidate_selection(pos=pos)
        # build the Multipartite graph and rank candidates using random
        # walk, alpha controls the weight adjustment mechanism
        extractor.candidate_weighting(alpha=1.1, threshold=0.74, method='average')
        # get the 10-highest scored candidates as keyphrases
        keyphrases = extractor.get_n_best(n=10)
        for val in keyphrases:
            out_list.append(val[0])
    except:
        out_list = []
        traceback.print_exc()
    return out_list
```

Results

Ex1:

```
['yellow', 'sun', 'orange', 'common misconception', 'red']
```

Ex2:

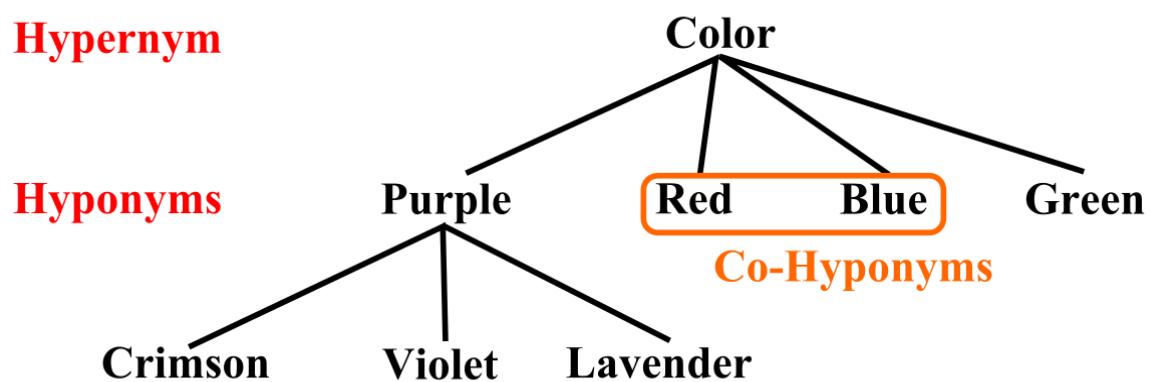
```
['java',
'computers',
'programming',
'popular programming language',
'secure',
'fast',
'free',
'data',
'powerful',
'world']
```

Find Distractors (Wrong Options) using Wordnet

Wordnet

WordNet is a large lexical database of English. Nouns, verbs, adjectives, and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. WordNet's structure makes it a useful tool for computational linguistics and natural language processing.

Distractors are co-hyponyms of the keyword



Implementation

```
# Get distractors for keywords using wordnet, return the keyword and list of distractors
def wordnet_distractors(keywords):
    distractors=[]
    for word in keywords:
        wr = word.lower()
        if len(wr.split())>0:
            # wordnet works with single words
            wr = wr.replace(" ", "_")
        # synsets of each word
        synset = wordnet.synsets(wr)
        for syn in synset:
            # hypernyms of each synset, work with only synsets that have hypernym
            hypernym = syn.hypernyms()
            if len(hypernym) == 0:
                continue
            # get hyponyms of the first hypernym
            for item in hypernym[0].hyponyms():
                # get root word of each hyponym
                name = item.lemmas()[0].name()
                # accept hyponyms which are not the original keyword
                if name == word:
                    continue
                # return hyponyms in a proper format
                name = name.replace("_", " ")
                if name is not None and name not in distractors:
                    distractors.append(name)
    # take first keyword in the list has distractors
    if len(distractors) > 0 :
        break
    return word, distractors
```

Results

Ex1:

```
Keyword: yellow
distractors: ['blond', 'blue', 'brown', 'complementary color', 'green', 'olive', 'orange', 'pastel', 'pink', 'purple', 'red', 'salmon', 'blacken', 'blush', 'dye'
```

Ex2:

```
Keyword: humans
distractors: ['actinoid', 'arrangement', 'association', 'biological group', 'circuit', 'citizenry', 'collection', 'community', 'edition', 'electron shell', 'ethr
Keyword: instructions
distractors: ['consuetudinary', 'grimoire', 'instruction manual', 'reference manual', 'sex manual', 'acknowledgment', 'approval', 'body', 'commitment', 'corker',
Keyword: world
distractors: ['asterism', 'black body', 'body', 'carpet', 'celestial body', 'cocoon', 'consolidation', 'constellation', 'covering', 'extraterrestrial object', 'i
Keyword: java
distractors: ['alcohol', 'cider', 'cocoa', 'coffee', 'cooler', 'drinking water', 'fizz', 'fruit drink', 'fruit juice', 'ginger beer', 'hydromel', 'mate', 'milk',
Keyword: secure
distractors: ['carry', 'derive', 'eke out', 'excavate', 'extort', 'extract', 'get in', 'grab', 'kite', 'mooch', 'procure', 'shop', 'snag', 'source', 'take out',
Keyword: java
distractors: ['alcohol', 'cider', 'cocoa', 'coffee', 'cooler', 'drinking water', 'fizz', 'fruit drink', 'fruit juice', 'ginger beer', 'hydromel', 'mate', 'milk',
Keyword: programs
distractors: ['burden', 'cogitation', 'concept', 'figment', 'generalization', 'ideal', 'idealization', 'impression', 'inspiration', 'keynote', 'kink', 'meaning',
Keyword: computer systems
distractors: ['audio system', 'communication system', 'computer system', 'containment', 'control system', 'data system', 'drainage system', 'exhaust', 'explosive
Keyword: data
distractors: ['agglomeration', 'ana', 'armamentarium', 'art collection', 'Asia', 'assortment', 'aviation', 'backlog', 'batch', 'battery', 'biota', 'block', 'book'
```

Filter Distractors using SciPy

SciPy

SciPy is open-source software for mathematics, science, and engineering. SciPy is a collection of mathematical algorithms and convenience functions built on the NumPy extension of Python. It adds significant power to the interactive Python session by providing the user with high-level commands and classes for manipulating and visualising data.

Theory

`Scipy.spatial.distance.cdist` computes the distance between each pair of the two collections of inputs using any of the distance metrics. We used Cosine Similarity to compute the distance between two vectors. To convert distractor and keyword texts to vector numbers, we use a sentence transformer to get sentence embeddings.

Implementation

```
# framework for state-of-the-art sentence, text and image embeddings.
from sentence_transformers import SentenceTransformer
# Choose the proper model for sentence embeddings https://www.sbert.net/docs/pretrained\_models.html
sent_model = SentenceTransformer('sentence-transformers/all-mpnet-base-v2')

# build on numpy to solve scientific and mathematical problems
import scipy
# get distractors and keyword embeddings
distractors_embeddings = sent_model.encode(distractors)
key_embedding = sent_model.encode([key])

# Compute distance between each pair of the two collections of inputs.
distances = scipy.spatial.distance.cdist(key_embedding, distractors_embeddings, 'cosine')[0]
```

Results

Ex1:

```
keyword: yellow
distractors: ['blond', 'blue', 'brown', 'complementary color', 'green', 'olive', 'orange', 'pastel', 'pink', 'purple', 'red', 'salmon', 'blacken', 'blush', 'dye'
distances: [0.40812856 0.26906401 0.34971107 0.48083187 0.26256597 0.62395227
 0.30378055 0.58434197 0.30950083 0.27713703 0.25917271 0.68198356
 0.50211143 0.71620907 0.45325517 0.30777827 0.4352093 0.49968218
 0.39738968 0.77357087 0.36612994 0.77448874 0.81464514 0.54943018]
```

Ex2:

```
keyword: data
distractors: ['agglomeration', 'ana', 'armamentarium', 'art collection', 'Asia', 'assortment', 'aviation', 'backlog', 'batch', 'battery', 'biota', 'block', 'boot
distances: [0.84678627 0.79129847 0.84393433 0.81128092 0.8498027 0.82219272
 0.83037359 0.80589928 0.797841 0.75077864 0.75641498 0.77098142
 0.83281654 0.81194593 0.8363447 0.95010107 0.72038688 0.868968
 0.83737551 0.84606785 0.85360921 0.65715634 0.86048602 0.66208533
 0.88486125 0.82814162 0.80136235 0.78548368 0.71918732 0.78306302
 0.63495672 0.82105637 0.80445484 0.94985916 0.84992067 0.84750648
 0.8954281 0.83260491 0.84592196 0.86754325 0.77825995 0.8093838
 0.79750308 0.79111178 0.79285222 0.7497867 0.86433572 0.8205502
 0.78622589 0.84309263 0.77988313 0.9015606 0.9332264 0.83407768
 0.80499497 0.7814374 0.85983866 0.86650521 0.86480046 0.78099454
 0.88087595 0.75568792 0.82474329 0.84967915 0.75835228 0.85036393
 0.88712418 0.77726428 0.74647288 0.88558288 0.9409595 0.86674319
 0.67915683 0.76203757 0.84832001 0.84040661 0.84680512 0.63657747
 0.77066916 0.7909172 0.81107886 0.83417868 0.79624977 0.86326957
 0.76296861 0.8108395 0.76996905 0.73801139 0.59471771 0.82008566
 0.47073776 0.7229532 0.63341889 0.74279513 0.74065544 0.85221422]
```

Final Output

Implementation

```
# first three distractors using minimum similarity distance, return list of three wrong options
def options_gen(distractors,distances,key):
    opts = []
    num = 5
    try:
        for i in range(num):
            # get index of minimum distance to take its corresponding distractor
            index, = np.where(distances == min(distances))
            dist = distractors[index[0]]
            # not accept replicated distractor even it with s or without
            if (key+'s' == dist) or (key == dist+'s') or (dist in opts):
                # delete the minimum to repeat process in the seconde minimum and so on
                distances = np.delete(distances,index[0])
                continue
            # add dist if it not in options
            opts.append(dist)
            distances = np.delete(distances,index[0])

    except:
        print("No Options")

    return opts[:3]
```

Results

Ex1:

Question: It is a common misconception that the Sun is _____, or orange or even red.
Answer: yellow
Wrong options ['red', 'green', 'blue']

Ex2:

Question: The _____ used to fit the model is called the training data
Answer: data
Wrong options ['evidence', 'datum', 'findings']

T/F Quiz

Ways to have a **True** sentence:

1. Tokenize a text into sentences and leave the sentence as is
2. Split the compound or complex sentence into a simple sentence

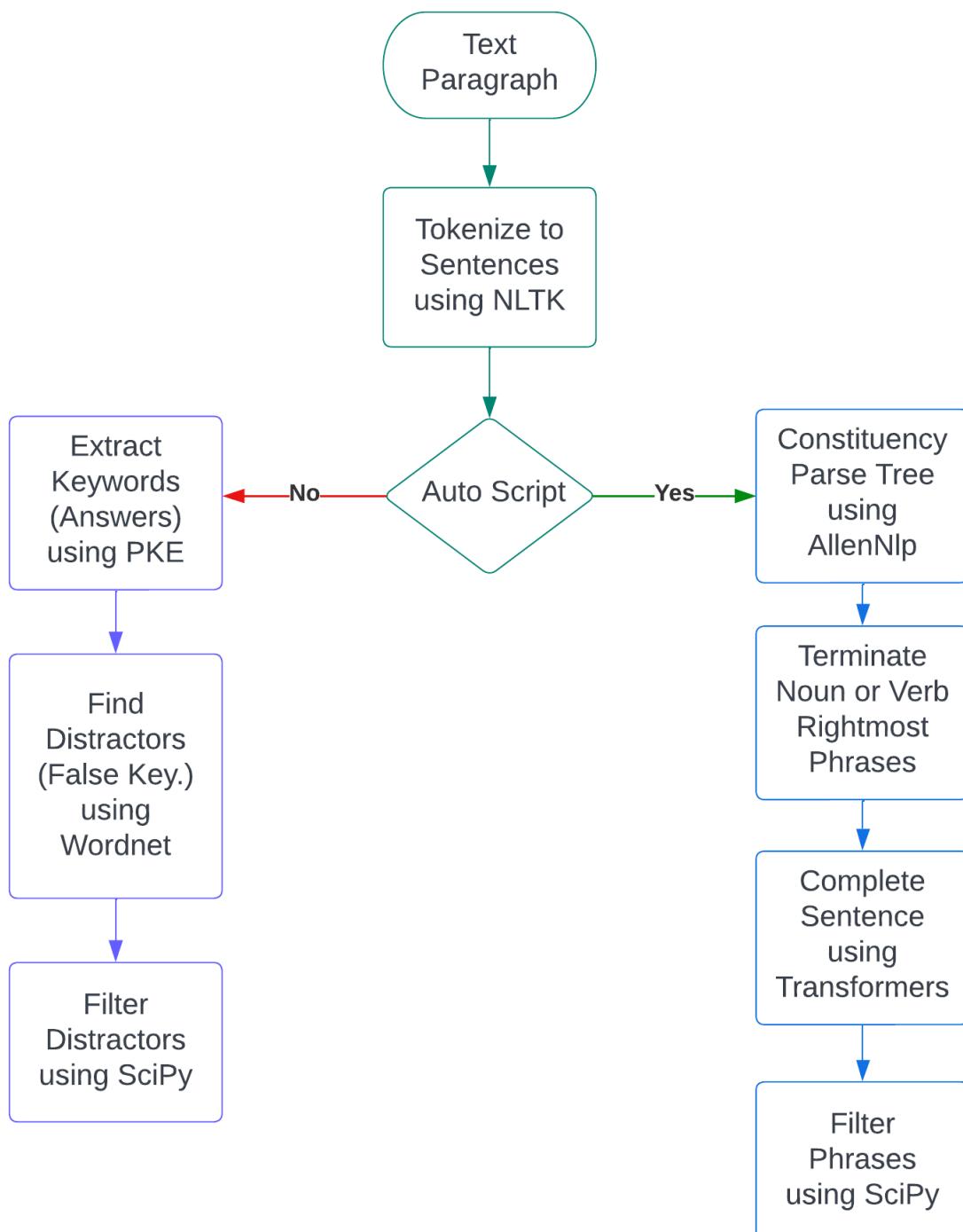
Ways to have a **False** sentence:

1. Add or remove negation of a verb phrase or noun phrase
2. Change a named entity
3. Change adjective
4. Change the main verb
5. Alternate verb phrase or noun phrase

To change entity, adjective, or verb in a sentence we use the same approach to get MCQ in the NLP model but instead of three distractors we get one and replace the answer with this distractor

But to alternate a verb phrase or a noun phrase we created a script that combined in it NLP techniques and the use of transformers and called it auto script

False Statements



AllenNLP

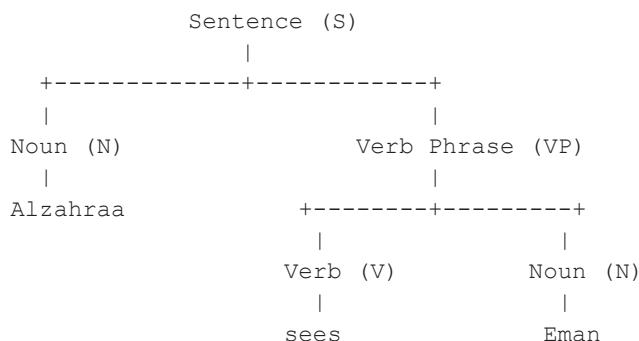
AllenNLP is a complete platform for solving natural language processing tasks in PyTorch.

Constituency Parsing

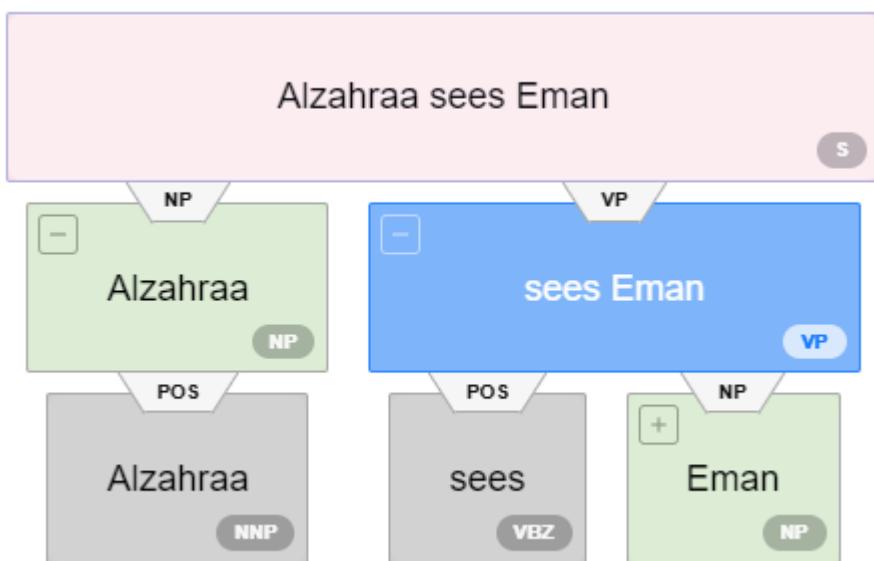
Constituency parsing is the task of breaking a text into sub-phrases or constituents. Non-terminals in the parse tree are types of phrases, the terminals are the words in the sentence.

Constituency parsing aims to extract a constituency-based parse tree from a sentence that represents its syntactic structure according to a phrase structure grammar.

Ex:



Allennlp provides this parse tree through its predictor model



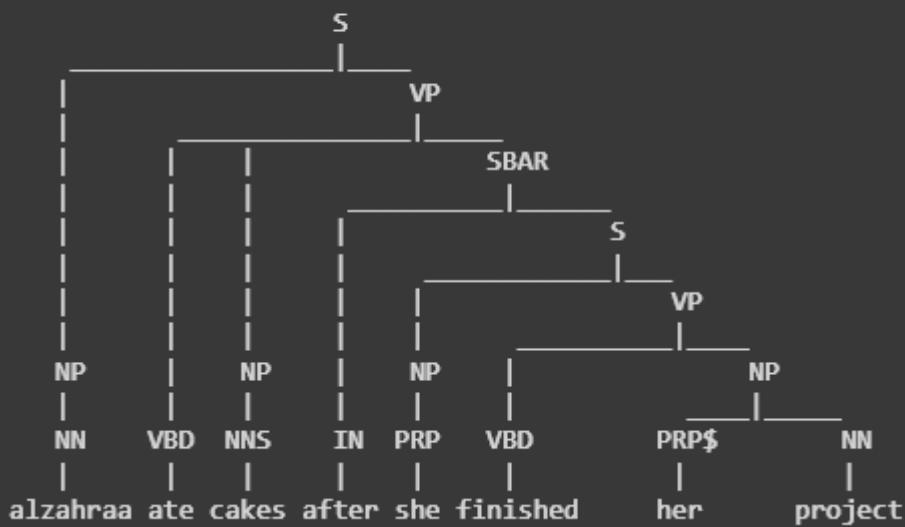
Result of Tree

```
test = "alzahraa ate cakes after she finished her project"

# remove any ending marks
test_sentence = test.rstrip('?:!.,;')
# pass sent into predictor function from allennlp
parsed_sent = predictor.predict(sentence=test_sentence)
tree_string = parsed_sent["trees"]
tree_string

'(S (NP (NN alzahraa)) (VP (VBD ate) (NP (NNS cakes)) (SBAR (IN after) (S (NP (PRP she)) (VP (VBD finished) (NP (PRP$ her) (NN project)))))))'
```

```
# convert tree into NLTK tree object
tree = Tree.fromstring(tree_string)
tree.pretty_print()
```



Result of Terminate Noun Phrases or Verb Phrases Rightmost

Get the rightmost (last) subtree if it is “NP” noun phrase or “VP” verb phrase and the function is done recursively to get the last phrases. We decide if we split NP or VP depending on the length, we take the longest

```
Ending phrase: finished her project
Original sentence : alzahraa ate cakes after she finished her project
Original sentence after splitting at ending phrase: alzahraa ate cakes after she
```

Result of Complete Sentence using Transformer

We used GPT-2 as Decoder Transformer to generate a completion of the sentence

```

generated_sentences=[]

for i, sample_output in enumerate(sample_outputs):
    decoded_sentence = GPT2tokenizer.decode(sample_output, skip_special_tokens=True)

    final_sentence = tokenize.sent_tokenize(decoded_sentence)[0]
    generated_sentences.append(final_sentence)
    print ("Output",i+1,": ",final_sentence)

Output 1 : alzahraa ate cakes after she had finished her day's work.
Output 2 : alzahraa ate cakes after she was stabbed, and her brother's mother said that when the attackers attacked on Wednesday
Output 3 : alzahraa ate cakes after she heard the news.
Output 4 : alzahraa ate cakes after she went to sleep, but then suddenly remembered that her husband was not in jail.
Output 5 : alzahraa ate cakes after she left to take a shower.

```

Result of Filter Phrases using SciPy

We used cosine similarity and sort phrases by most similar to the original sentence.

```

[(0, 0.06214037577395015), (2, 0.18716594922705843), (4, 0.1967469118294214), (1, 0.2599666092291667), (3, 0.2841115474934752)]

dissimilar_sentences =[]
for idx, distance in results:
    dissimilar_sentences.append(generated_sentences[idx])
    print (generated_sentences[idx])

alzahraa ate cakes after she had finished her day's work.
alzahraa ate cakes after she heard the news.
alzahraa ate cakes after she left to take a shower.
alzahraa ate cakes after she was stabbed, and her brother's mother said that when the attackers attacked on Wednesday
alzahraa ate cakes after she went to sleep, but then suddenly remembered that her husband was not in jail.

```

Assignment Reviewer and OCR Reader

Tools and software:

JupyterLab (Notebook)

Free software, open standards, and web services for interactive computing across all programming languages.

Google collaboratory (Colab)

Google Colab is an excellent tool for deep learning tasks. It is a hosted Jupyter notebook that requires no setup and has an excellent free version, which gives free access to Google computing resources such as GPUs and TPUs. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education.

OCR Reader

For reading pdf files using OCR part:

We have used this particular model after several attempts using different traditional commands to read files such as (Tika-parser, reading word, reading powerpoint, ..)

The biggest challenge for reading the pdf files is that the file can be based on images and not its original source as a text or a word file, therefore the performance of the traditional commands was not good, so the OCR model was used instead of it due to its quality in extracting the text and reading the files regardless of their original source.

A model was used pytesseract instead of easy-ocr because it showed more accuracy during a shorter execution period with large texts. The easy-ocr is used

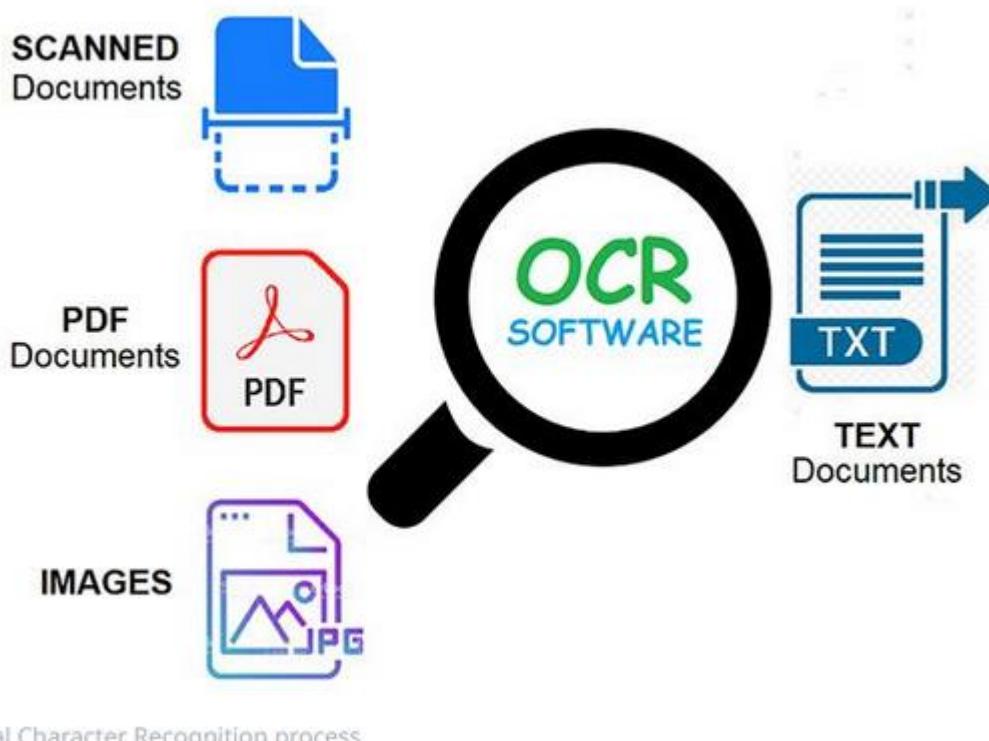
with numbers, especially in radars, to take small texts such as car plates. As for large texts, the best in comparison is a model pytesseract.

Methods and Modules:

Tesseract-ocr (1)

Tesseract is an optical character recognition engine for various operating systems. It is free software, released under the Apache Licence.

In other words, OCR systems transform a two-dimensional image of text, that could contain machine printed or handwritten text from its image representation into machine-readable text. OCR as a process generally consists of several sub-processes to perform as accurately as possible.



Pdf2image (2)

A python module that converts PDF to Image object.

Poppler-utils (3)

Precompiled command-line utilities (based on Poppler) for manipulating PDF files and converting them to other formats.

Pillow (PIL) (4)

Pillow is Python Imaging Library

The Python Imaging Library adds image processing capabilities to your Python interpreter.

This library provides extensive file format support, an efficient internal representation, and fairly powerful image processing capabilities.

The core image library is designed for fast access to data stored in a few basic pixel formats. It should provide a solid foundation for a general image processing tool.

Clean-text (5)

cleantext is an open-source python package to clean raw text data.

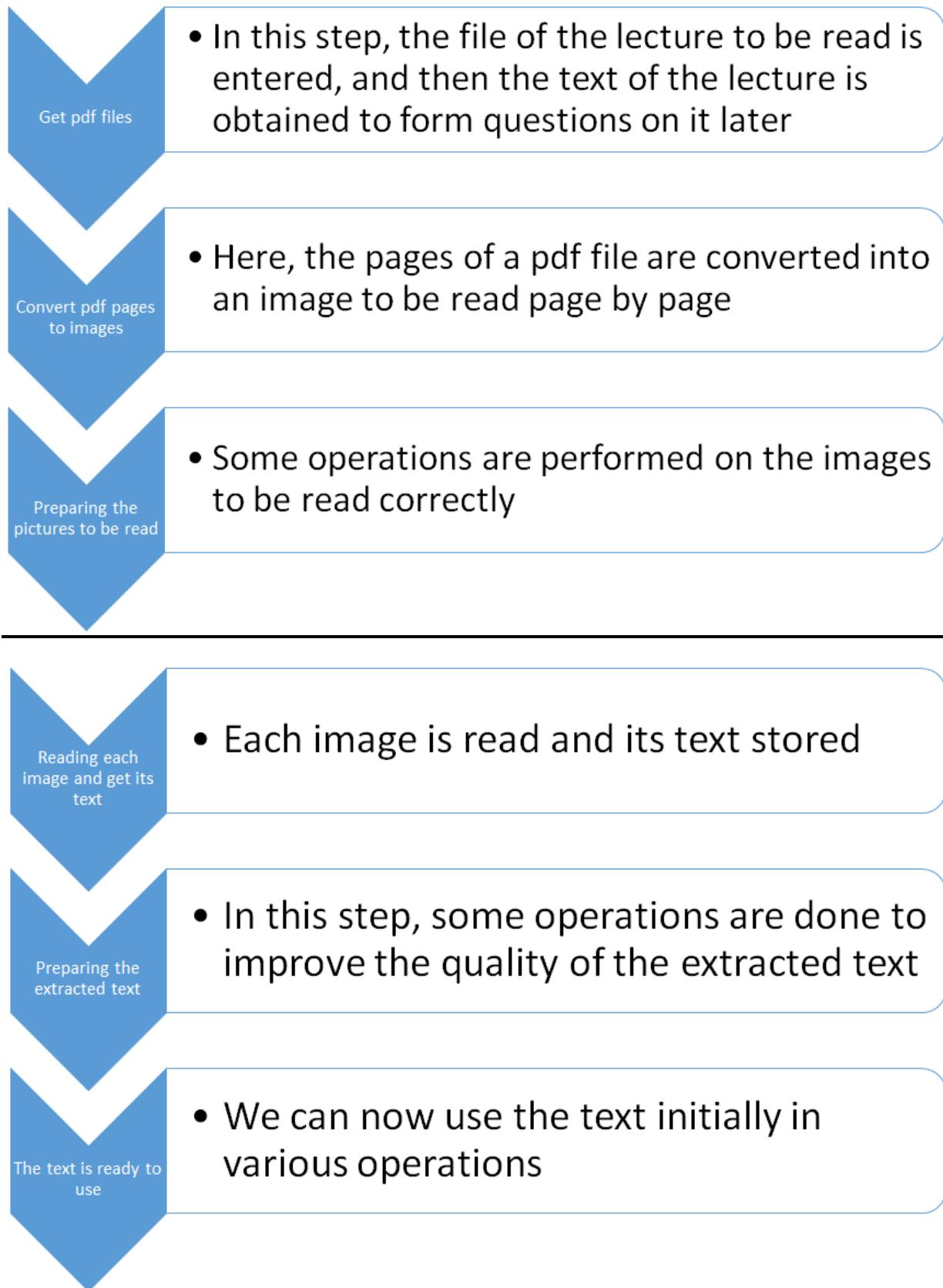
Preprocess your scraped data with clean-text to create a normalised text representation.

cv2 (open cv) (6)

OpenCV-Python is a library of Python bindings designed to solve computer vision problems.

cv2.imread() method loads an image from the specified file. If the image cannot be read (because of missing file, improper permissions, unsupported or invalid format) then this method returns an empty matrix.

The steps to implement the task



Code implement :

1- Install requirements

Install libraries and import them in the code.

```
!sudo apt install tesseract-ocr 4.0.0
!sudo apt-get install tesseract-ocr-eng
!sudo apt-get install tesseract-ocr-ara
!pip install pytesseract
!pip install pdf2image
!apt-get install poppler-utils
!pip install PIL
!pip install Pillow 4.0.0
!pip install cleantext
```

2- Get pdf files

In this step, the file of the lecture to be read is entered, and then the text of the lecture is obtained to form questions on it later

3- Convert pdf pages to images

Here, the pages of a pdf file are converted into an image to be read page by page and determine the properties of the saved image.

```
# Convert PDF_pages to Images
pages = convert_from_path(pdf_file, dpi=200,fmt='jpeg')
```

4- Preparing the pictures to be read

Some operations are performed on the images to be read correctly, such as converting to grayscale to increase the quality of the image and remove the noise and determine the threshold for good quality.

```

# Convert image to grayscale image
def get_grayscale(image):
    return cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Remove noise from image
def remove_noise(image):
    return cv2.medianBlur(image,5)

# Get thresholding image
def thresholding(image):
    return cv2.threshold(image, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)[1]

```

5- Reading each image and get its text

Each image is read and its text stored using pytesseract.image_to_string() function.

```

# Using pytesseract to extract the text from stored image
ocr_result_original= pytesseract.image_to_string(final_image,config=custom_config)

```

6- Preparing the extracted text

In this step, some operations are done to improve the quality of the extracted text using the cleantext.text() , to convert the text to lowercase and delete extra spaces.

```

# To clean the extracted text
#clean text will be saved in database to generate the question later
clean_text = cleantext.clean(final_txt, # text to be cleaned
clean_all=False, # Execute all cleaning operations
extra_spaces=True, # Remove extra white spaces
lowercase=True, # Convert to lowercase
stp_lang='english' # Language for stop words
)

```

7- The text is ready to use

Now, we can now use the text initially in various operations.

Results

The result of the model is the text of lecture file (pdf file)

Enter path of lecture file: /content/01_Introduction_to_PC.pdf
Introduction

CSE 425 Process Control
Lecture 1

Dr. Ahmed Alenany
About the course
Grading Scheme

Semester work

Final
Total

References

1. R.Rengaswamy, B. Srinivasan, and N. P. Bhatt, Process Control Fundamentals-Analysis, Design, Assessment, and Diagnosis, CRC Press, 2020.
2. H. Wade, Basic and Advanced Regulatory Control: System Design and Application, 2" edition, 2004.
3. T. Blevins and M. Nixon, Control Loop Foundation Batch and Continuous Processes, ISA, 2011. 5

Topics to be covered:

✓ 48s completed at 9:43 AM

Multi-loop interaction

Process

Operation (chemical or physical) used to convert feed materials into products.

Examples: mixing - separation - heating - cooling - filtering - compression - etc.

The term process is used for both the processing operation and the processing equipment.

Nearly every product we use today is the output of some process industry (Food – Pharmaceuticals – Paper – Rubber – Oil and Gas – Textiles – Glass – Steel)

The design of such processes are part of process engineering (under chemical engineering)
Process Control

Methods used to control Process Variables (PV) when manufacturing a product.

PV examples: Temperature – Pressure - Flow - Concentration of chemicals - Liquid levels

Process control maintains a PV at the desired operating conditions despite any disturbance. This is achieved by

✓ 48s completed at 9:43 AM

Notes

The operating time varies between each file and the other according to the file size and quality.

Assignment Reviewer

Grading Assignment

For grading programing assignment part:

A subprocess function has been used in order to be able to deal with the command window (cmd) and thus we can make compile to the code file and then we can grade it by measuring the percentage of similarity (using sequencematcher function) between the student's answer after running his code file and the code file of the doctor, which represents the answer model for the assignment and through it the percentage of similarity between the two outputs can be attributed to a specific number determined by the doctor to be the final score for the test, and thus the score of each student is calculated

Methods and Modules:

Subprocess (7) (Subprocess management) method

The subprocess module allows you to spawn new processes, connect to their input/output/error pipes, and obtain their return codes. This module intends to replace several older modules and functions.

SequenceMatcher (8)

SequenceMatcher is a class available in a python module named “difflib” - this module provides classes and functions for comparing sequence deltas - It can be used for comparing pairs of input sequences. The objective of this article is to explain the SequenceMatcher algorithm through an illustrative example. Due to the limited docs available, I thought to share the concept through a step by step example which can help the reader in understanding the entire procedure in a lucid manner.

The basic idea is to find the longest contiguous matching subsequence (LCS) that contains no “junk” elements. This does not yield minimal edit sequences, but does tend to yield matches that “look right” to people.

Example about sequence matcher:

Input strings:

A: THANKS FOR RESPONSE

B: THANKING FOR KIND RESPONSE

A:

T	H	A	N	K	S		F	O	R		R	E	S	P	O	N	S	E
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

B:

T	H	A	N	K	I	N	G		F	O	R		K	I	N	D		R	E	S	P	O	N	S	E
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

The matching blocks after one iteration:

Longest matching: **RESPONSE**

	R	E	S	P	O	N	S	E
10	11	12	13	14	15	16	17	18

The strings:

A:

T	H	A	N	K	S		F	O	R
0	1	2	3	4	5	6	7	8	9

B:

T	H	A	N	K	I	N	G		F	O	R		K	I	N	D
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

The matching blocks after second iteration:

Matching: **RESPONSE** and **THANK**

	R	E	S	P	O	N	S	E
10	11	12	13	14	15	16	17	18

T	H	A		N	K
0	1	2		3	4

The strings:

A:

S		F	O	R
5	6	7	8	9

B:

I	N	G		F	O	R		K	I	N	D
5	6	7	8	9	10	11	12	13	14	15	16

The matching blocks after third iteration:

Matching: **RESPONSE** and **THANK** and **FOR**

	R	E	S	P	O	N	S	E
10	11	12	13	14	15	16	17	18

T	H	A	N	K
0	1	2	3	4

	F	O	R
6	7	8	9

The strings:

A:

S
5

B:

I	N	G		K	I	N	D
5	6	7	12	13	14	15	16

Similarity Ratio:

Ratio = (2*N)/ Total length of both sequence; N = Number of characters matched

$$= (2*18)/45$$

$$= 0.8$$

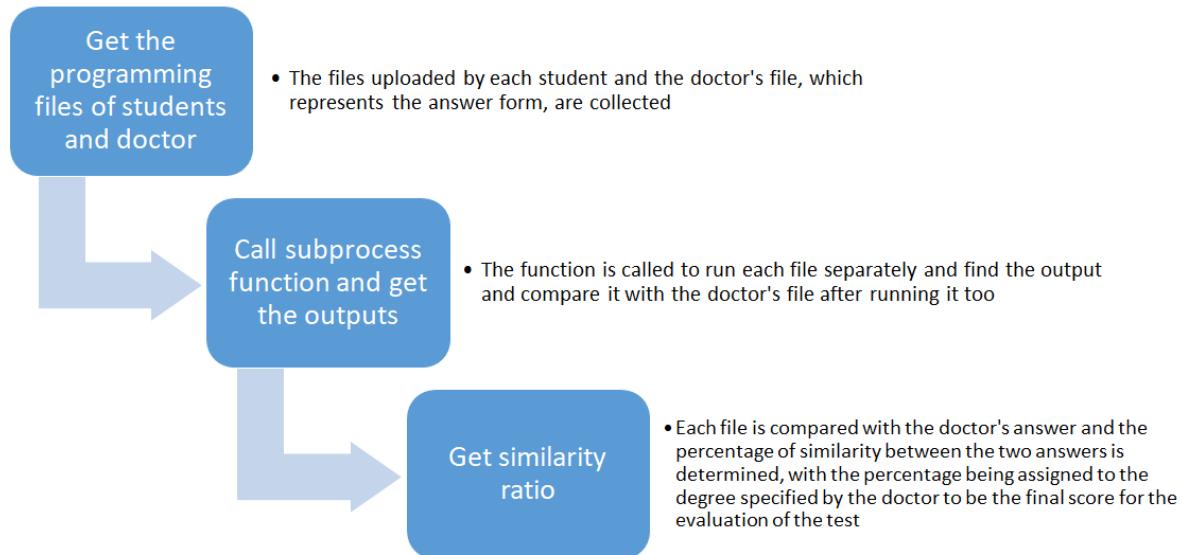
```

>>> from difflib import SequenceMatcher
>>> seq = SequenceMatcher(None, "THANKS FOR RESPONSE" , "THANKING FOR KIND RESPONSE")
>>> list(seq.get_matching_blocks())
[Match(a=0, b=0, size=5), Match(a=6, b=8, size=4), Match(a=10, b=17, size=9), Match(a=19, b=26, size=0)]
>>> seq.ratio()
0.8

```

SequenceMatcher (fig 1)

The steps to implement the task



Code implement :

1- Install requirements

Install libraries and import them in the code.

```

pip install subprocess
pip install os
from difflib import SequenceMatcher as sm

```

2- Get the programming files of students and doctor

The files uploaded by each student and the doctor's file, which represents the answer form, are collected

3- Call subprocess function and get the outputs

The function is called to run each file separately and find the output and compare it with the doctor's file after running it too.

```

def executeJavaDoctor():
    global Java_Output_doctor
    s = subprocess.getoutput("javac HelloWorldDoctor.java;java HelloWorldDoctor")
    Java_Output_doctor=s
    print("Java_Output_doctor is : " + Java_Output_doctor)

def executeJavaStudent():
    global Java_Output_student
    s = subprocess.getoutput("javac HelloWorldStudent.java;java HelloWorldStudent")
    Java_Output_student=s
    print("Java_Output_student is : " + Java_Output_student)

```

4- Get similarity ratio

Each file is compared with the doctor's answer and the percentage of similarity between the two answers is determined, with the percentage being assigned to the degree specified by the doctor to be the final score for the evaluation of the test.

```

similarity=sm(None,Java_Output_doctor,Java_Output_student).ratio()
SetAssignmentDegree=input("Enter the assignment grade")
Final_degree=similarity*int(SetAssignmentDegree)
print(Final_degree)

```

Results

The result of the model is the degree of the student after grade the his assignment answer.

```

Java_Output_doctor is : My name is Java Programming Language
Java_Output_student is : My name is Java programming language,my name is marco
Enter the assignment grade: 5
3.8202247191011236

```

```

Java_Output_doctor is : My name is Java Programming Language
Java_Output_student is : My name is Java Programming Language
Enter the assignment grade: 5
5.0

```

Notes

The professor can set the assignment score and the model will correct the students' files according to the professor's specified score.

Plagiarism Detector

For plagiarism detector for programming assignments part:

After correcting student files, this model can find similar solutions (Plagiarism ratio detector) between students and find the percentage of Plagiarism between them by comparing the code text (syntax) for each student and the other.

Methods and Modules:

Scikit-learn (9)

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modelling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python.

TfidfVectorizer (10)

TfidfVectorizer is Term Frequency Inverse Document Frequency. This is a very common algorithm to transform text into a meaningful ratio or number.

In TfidfVectorizer we consider the overall document weightage of a word. It helps us in dealing with the most frequent words. Using it we can penalise them.

TfidfVectorizer weights the word counts by a measure of how often they appear in the documents.

Tf-idf value of a term in a document is the product of its tf and idf where : (tf) is the number of times a term appears in a particular document, and (idf) is a measure of how common or rare a term is across the entire corpus of documents.

The higher the value, the more relevant the term is in that document.

Count Vectorizer

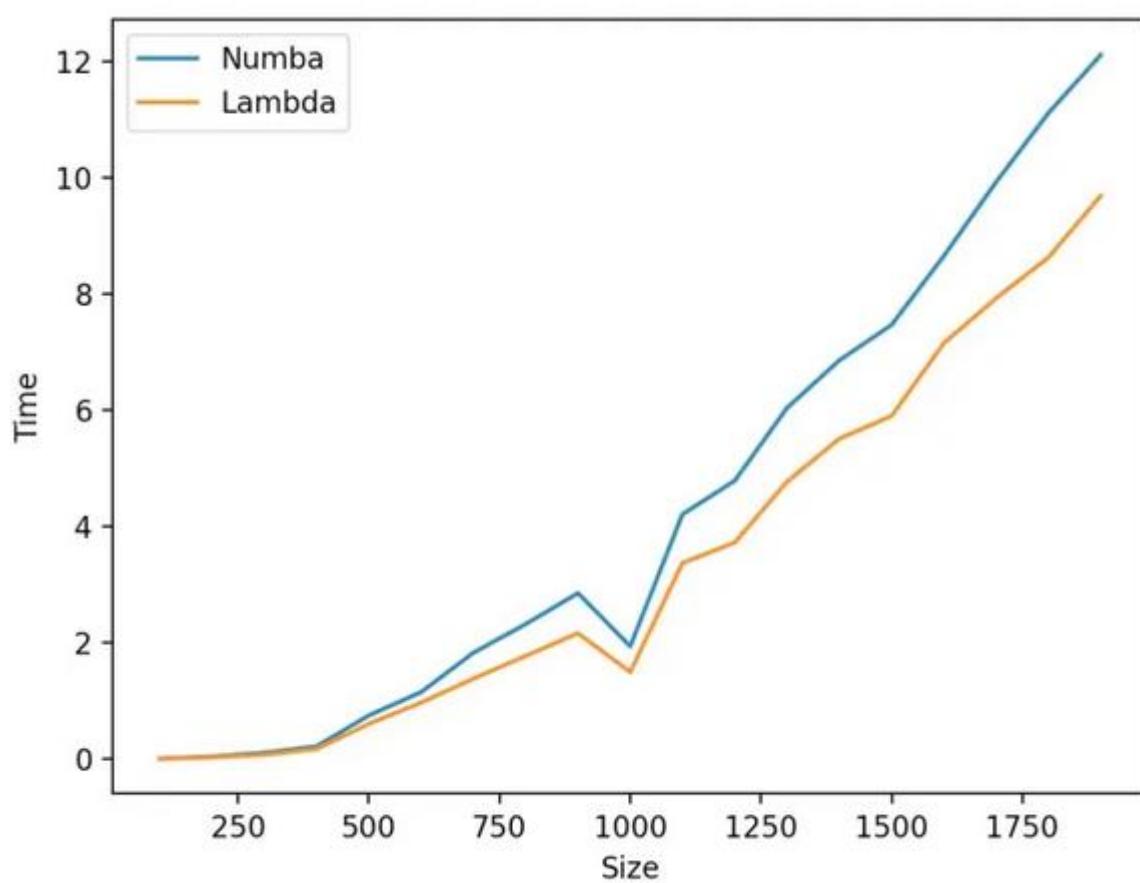
	blue	bright	sky	sun
Doc1	1	0	1	0
Doc2	0	1	0	1

TD-IDF Vectorizer

	blue	bright	sky	sun
Doc1	0.707107	0.000000	0.707107	0.000000
Doc2	0.000000	0.707107	0.000000	0.707107

TF-IDF Vectorizer scikit-learn (fig 1)

We use Lambda as it has higher performance than Numba (11)

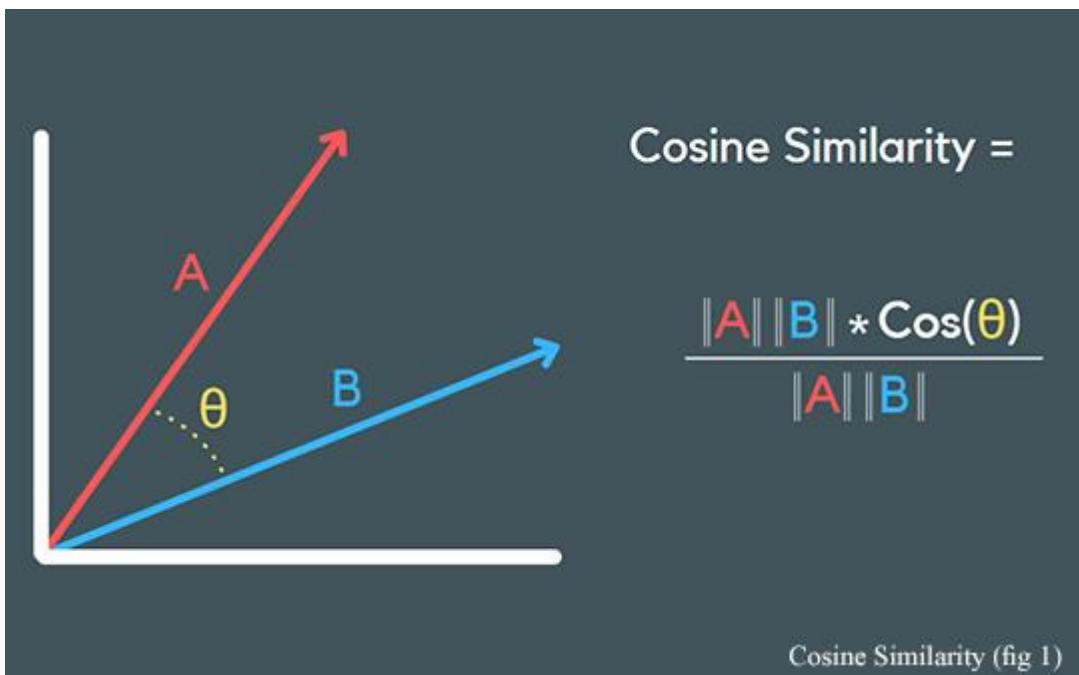


Performance comparison Numba VS Lambda (fig 1)

Cosine similarity

Cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them.

Cosine similarity, or the cosine kernel, computes similarity as the normalised dot product of A and B.



Cosine Similarity (fig 1)

The steps to implement the task

Get the syntax of all programming files

Read all student files and get the code text (code syntax).

Convert syntax to array and get the similarity

The text is converted into numbers inside an array then we measure the similarity ratio between files

Get similar files according to the percentage

All two similar files are found, whose similarity ratio exceeded the allowed percentage

Code implement :

1- Install requirements

Install libraries and import them in the code.

```
pip install -U scikit-learn  
pip install os  
import os  
from sklearn.feature_extraction.text import TfidfVectorizer  
from sklearn.metrics.pairwise import cosine_similarity
```

2- Get the syntax of all programming files

Read all student files and get the code text (code syntax).

```
student_files = [doc for doc in os.listdir() if doc.endswith('.java')]  
student_notes =[open(File).read() for File in student_files]
```

3- Convert syntax to array and get the similarity between files

The text is converted into numbers inside an array using a function TfidfVectorizer() and then Cosine_similarity function is used to measure the similarity ratio between files by multiplying the matrix of each file with the matrix of the other file.

```
vectorize = lambda Text: TfidfVectorizer().fit_transform(Text).toarray()  
similarity = lambda doc1, doc2: cosine_similarity([doc1, doc2])
```

4- Get similar files according to the percentage

In this step, all two similar files are found, whose similarity ratio exceeds the permissible percentage (allowed ratio) determined by the professor of the subject.

```
Allowed_similarity_ratio = input("Enter the allowed similarity ratio:")
def check_plagiarism():
    plagiarism_results = set()
    global s_vectors
    for student_a, text_vector_a in s_vectors:
        new_vectors = s_vectors.copy()
        current_index = new_vectors.index((student_a, text_vector_a))
        del new_vectors[current_index]
        for student_b , text_vector_b in new_vectors:
            sim_score = (similarity(text_vector_a, text_vector_b)[0][1])*100
            if sim_score > eval(Allowed_similarity_ratio) :
                student_pair = sorted((student_a, student_b))
                score = (student_pair[0], student_pair[1],sim_score)
                plagiarism_results.add(score)
    return plagiarism_results
```

Results

The result of the model is the similar files in the code syntax between the students and the percentage of similarity between each of them

```
['FibonacciStudent.java', 'HelloWorldDoctor.java', 'HelloWorldStudent1.java', 'Smiley.java', 'SpiralPatternExampleDoctor.java',  
 ['class FibonacciStudent {\n    public static void main(String[] args) {\n        int n = 200, firstTerm = 0, secondTerm = 1;\n        System.out.print(firstTerm + " " + secondTerm);  
        for (int i = 2; i < n; i++) {\n            int nextTerm = firstTerm + secondTerm;  
            System.out.print(" " + nextTerm);  
            firstTerm = secondTerm;  
            secondTerm = nextTerm;  
        }  
    }  
}']
```

```
Enter the allowed similarity ratio:50
('SpiralPatternExampleDoctor.java', 'SpiralPatternExampleStudent.java', 99.90212514385843)
('FibonacciDoctor.java', 'FibonacciStudent.java', 95.49555551835984)
('HelloWorldStudent.java', 'HelloWorldStudent1.java', 81.19948081736784)
('HelloWorldDoctor.java', 'HelloWorldStudent1.java', 73.66017161387155)
('HelloWorldDoctor.java', 'HelloWorldStudent.java', 83.5993448215934)
```

```
Enter the allowed similarity ratio:80
('HelloWorldStudent.java', 'HelloWorldStudent1.java', 81.19948081736784)
('SpiralPatternExampleDoctor.java', 'SpiralPatternExampleStudent.java', 99.90212514385843)
('FibonacciDoctor.java', 'FibonacciStudent.java', 95.49555551835984)
('HelloWorldDoctor.java', 'HelloWorldStudent.java', 83.59934482159339)
```

```
Enter the allowed similarity ratio:90
('SpiralPatternExampleDoctor.java', 'SpiralPatternExampleStudent.java', 99.90212514385843)
('FibonacciDoctor.java', 'FibonacciStudent.java', 95.49555551835984)
```

Notes

The professor can set the allowed plagiarism ratio and the model will get similar files and the similarity ratio between them.

Chatbot:

- **AI Virtual Assistant:**

Everyday life has become smarter and more intertwined with technology in our modern era. We are already familiar with chat assistants such as (Google, Siri), etc. It may now function as a basic medical prescriber, daily calendar reminder, note writer, calculator, and search tool in our chatbot assistance system.

Assistants are software applications that assist you with day-to-day duties such as displaying weather information, preparing remainders, making grocery lists, and so on.

They can respond to orders via text (as in online chatbots).

There are several virtual assistants available, including Apple's Siri, Amazon's Alexa, and Microsoft's Cortana.

Chatbots simulate a person-to-person chat to provide customer care help and link consumers with the services or information they require.

This project uses keyboard input to generate output and displays text on the screen.

The major goal of our chatbot help is to make students smarter by providing quick and calculated outcomes.

- **Materials:**

- **Watson Assistant: Intelligent virtual agent**

IBM's Watson Assistant employs artificial intelligence to give rapid, consistent, and accurate responses across any application, device, or channel. With the leader in trustworthy AI, you can say goodbye to long wait times, time-consuming searches, and useless chatbots.

We can repost a JavaScript snippet into the HTML of your current website using your assistant's Integrations page. This will generate a completely customizable web widget on the page we choose. Watson Assistant can also be integrated with Slack,

WhatsApp, Facebook Messenger, Amazon Alexa, social media platforms, and CRM systems. Our chatbot API, of course, interacts with other Watson APIs.

Technology that is used in chatbots:

Chatbots use natural language processing (NLP). Natural language processing makes it possible for our bot to read text, hear and interpret speech, measure sentiment, and determine which parts are important.

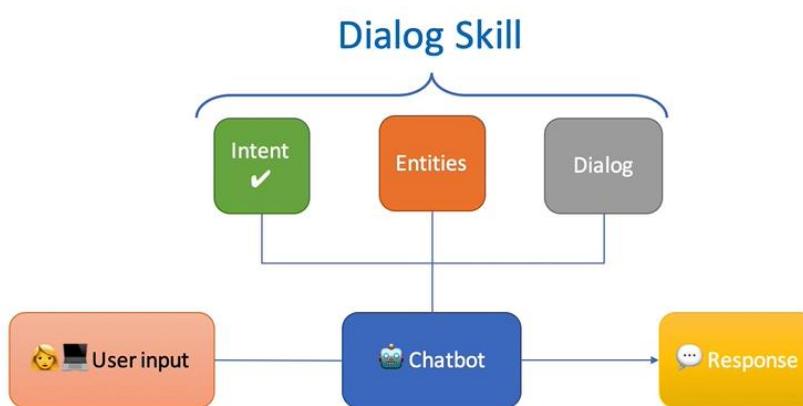
Chatbots can learn by analyzing the data sets we provide, and through the dialog with our users. Chatbots can also learn by having humans edit the system. In most cases, both are required.

Watson Assistant understands how to effectively answer end-user inquiries using machine learning and intent recognition methods.

Watson Assistant's artificial intelligence is meant to accurately understand the numerous variations of intent in real-world interactions.

A conversational flow

The component through which we develop the chatbot's communication with our users. The dialog allows the chatbot to establish user intent and deliver a pre-written answer, giving the user a clear sense of what the chatbot is there to perform.



Best online learning experience.

Chatbot Icon will always be shown at every page on the website as in the figure:

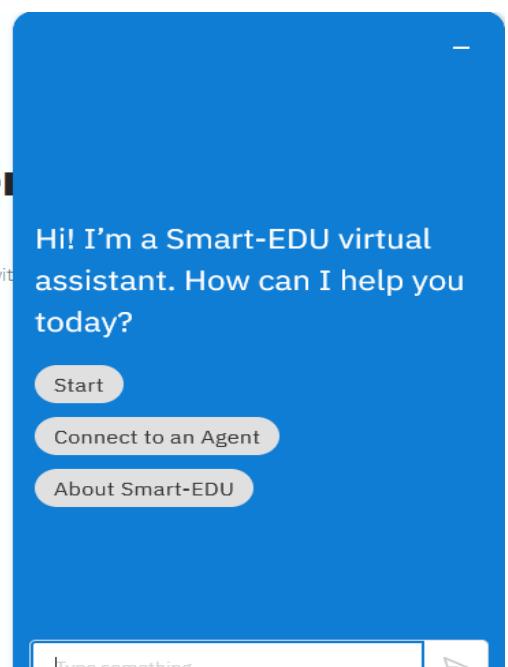
No more frequent flaws, no boundaries between the teacher and— students with a complete platform that works on all devices, you can't miss a thing!

[Read more about us](#)

X Close

Hi! I'm a virtual assistant.
How can I help you
today?

The opening screen is equipped to start use, as well as in the event that the student wants to talk to any assistant:



Best online learning experience.

No more frequent flaws, no boundaries between the teacher and— students with a complete platform that works on all devices, you can't miss a thing!

[Read more about us](#)

Welcome Screen:

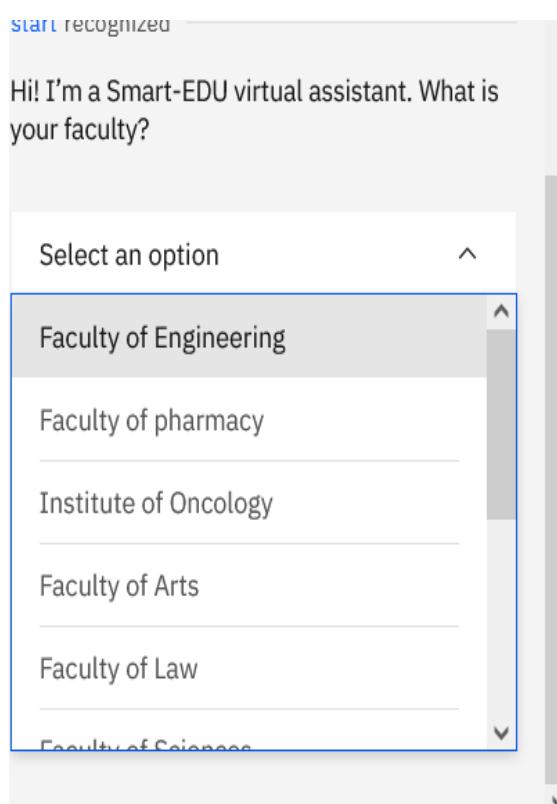
Welcome, how can I assist you?

start

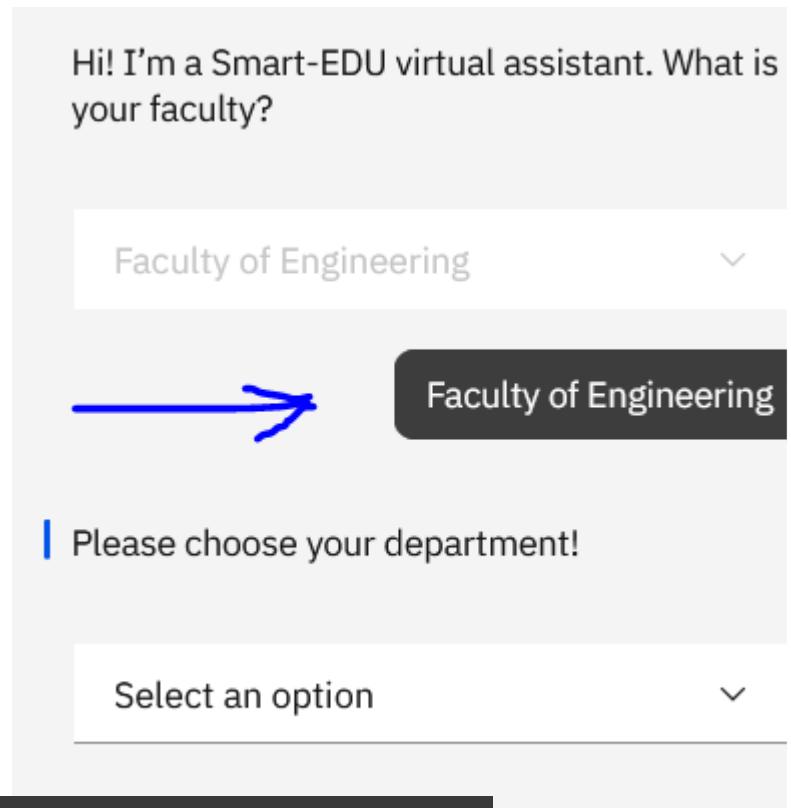
Page 198

```
1 {
2   "generic": [
3     {
4       "values": [
5         {
6           "text_expression": {
7             "concat": [
8               {
9                 "scalar": "Hi! I'm a Smart-EDU virtual
10                assistant. What is your faculty?"
11              }
12            ]
13          }
14        ],
15        "response_type": "text",
16        "selection_policy": "sequential"
17      },
18    ]
19 }
```

List of Faculties of Zagazig University to choose the student faculty:



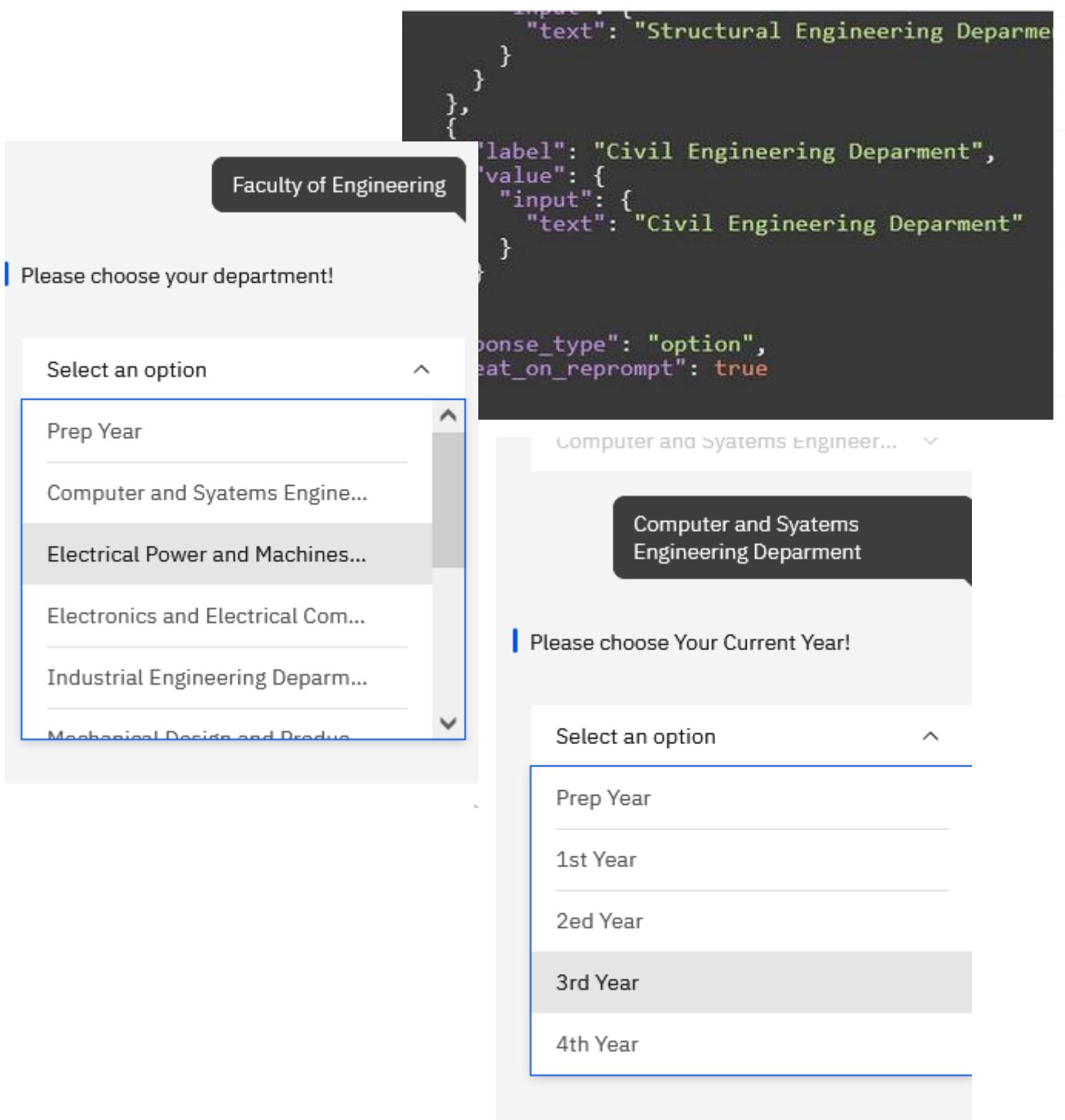
EX: If the student chooses Faculty of Engineering:



Here is the backEnd code:

```
generic": [
  {
    "values": [
      {
        "text_expression": {
          "concat": [
            {
              "scalar": "Please choose your department!"
            }
          ]
        }
      }
    ],
    "response_type": "text",
    "selection_policy": "sequential"
  },
  {
    "values": [
      {
        "text_expression": {
          "concat": [
            {
              "scalar": "Please choose your department!"
            }
          ]
        }
      }
    ],
    "response_type": "text",
    "selection_policy": "sequential"
  }
]
```

I Create a List of Our Engineering Departments as shown:



After the student choose his department and his Current year, Bot will ask him/her to choose Subject/Course as shown:

```
{  
  "g  
  ric": [  
    values": [  
      {  
        "text_expression": {  
          "concat": [  
            {  
              "scalar": "Please choose your course?"  
            }  
          ]  
        }  
      }  
    ],  
    response_type": "text",  
    selection_policy": "sequential"  
  ]  
}
```

Please choose your course?

Select an option ^

ARTIFICIAL INTELLIGENCE

Computer Network

Measurements&Labs 1st Term

DIGITAL CONTROL

DIGITAL COMMUNICATION SY...

```
{  
  response_type": "option",  
  repeat_on_reprompt": true,  
  options": [  
    {  
      "label": "ARTIFICIAL INTELLIGENCE",  
      "value": {  
        "input": {  
          "text": "ARTIFICIAL INTELLIGENCE"  
        }  
      }  
    },  
    {  
      "label": "Computer Network",  
      "value": {  
        "input": {  
          "text": "Computer Network"  
        }  
      }  
    }  
  ]  
}
```

We have designed this virtual assistant to be of assistance and extension for students when they need quick access to materials or quizzes, which consequently saves more time and effort for them.

More improvements will be made to this virtual assistant:

Establishing the rest of the departments of all other colleges, not just the College of Engineering.

Adding the subjects for each academic year in each of the remaining faculties.

Make the virtual assistant as integrated as possible.

Chapter 5

Conclusion and Recommendations

Strength:

This project contributed to solving many problems and facilitating many things for both the student and the lecturer (teacher).

For Teacher:

He has long suffered from the work of quizzes and assignments for the students, as well as from correcting them and the grades of all students and collecting grades for each student during the course to evaluate them, so we facilitated his work through the following:

1. The ease of raising the lecture video and the lecture file and the ease of communicating with the students through the comments below each lecture.
2. Not to open the next unit to the student until he/she has completed the unit in which he/she enables the student to master this unit and determine his/her level through his/her quizzes and assignments.
3. Easy to work through a quiz for students using AI in two ways:
 - ◆ can upload a file or enter the text that he wants to make questions about.
 - ◆ He can also determine the lecture he wants to do the quiz on, and AI makes quizzes for students. We just want to know when the quiz is going to be.
4. It's easy to add a task to each topic and identify its type as text or programming by selecting the language in it.
5. We have contributed to reducing fraud in two ways:
 - ◆ The first is to prevent the student from cheating during the quiz by warning him twice and the third is to cancel the quiz for him.
 - ◆ And the other is in the programming assignment correction, where we compared the student codes to each other to see if they cheated.
6. The level of each student in each course is determined by the leaderboard.

7. Can follow students through the course's chat rooms.
8. A meeting can be held at any time with students. This meeting is characterised by the use of the camera and microphone by both parties (student and teacher) and does not stop after a specified period of time.

All of this is in addition to normal things such as adding any number of files, quizzes, whether using AI or manual, and assignments for each topic in his course, and the possibility of adding more than one course to the same teacher.

For students:

We students have always suffered from not knowing the dates of the assignments and scheduling the quizzes accurately, in addition to not knowing the grades of any of them, and therefore not specifying at what level I am as a student, and the difficulty of communicating with the teacher, so we contributed to the following:

1. Easy scheduling of assignments and quizzes and knowing what to do throughout the course.
2. Increase students' desire to learn the course through the points calculation if they see the video or answer one of the other students' questions, which helps students help each other.
3. Knowledge of the student's grades at any time during the course increases the student's desire to increase his level (points)
4. The presence of all students' grades on the leaderboard, which helps the student to know their superiors at any time during the course, increases the competition between them.
5. Not to open the next unit to the student until he finishes the unit in which he enables the student to master this unit and determine his level through his grades in quizzes and assignments. The student can know in which topic he is weak so that he can return the video again or communicate with the lecturer or another student to understand it.
6. Easy communication with other students and lecturers through chat rooms.
7. Easy to achieve information at any time by using a chatbot.
8. Prevent cheating which helps the student to know his real level in the course and help him to raise his level.

Next Version

The main goal of every system update is to make the system more interactive with users and increase the independence to work in other words. to reduce dependence on individuals to manage the system and rely on artificial intelligence methods and machine learning, thus enhancing the wonderful experience for users, whether they are students and professors.

For professors, we can add features that facilitate the teaching process and follow the progress of students more accurately. Here are some points in future updates:

1- Self-correcting handwritten papers:

by using AI and ML we can deal with handwritten papers and answer assignments and read them using OCR technologies

2- Smart chatbot to help students study:

to answer students' questions about scientific content using an AI model that uses Transformers and deep learning.

3- Adding more languages to the programming assignments reviewing system:

We can review programming assignments in more programming languages in addition to the current language (Java), so we can add C or C++.

4- Display more reports and statistics:

using data analysis to clarify students' activities, know their progress, and follow up on the educational process.

5- Adding some tools to help professors such as notes and graphics software.

For students, the main goal is to make the learning process and its organization easier and more attractive than the traditional way of teaching by providing interesting and attractive ways of explanation, which motivate students to study while creating competitive opportunities among students. Here are some points in future updates:

1- Possibility of conducting competitions among students:

It is possible to conduct competitions on competitive projects among students to increase the practical application of the curricula and the parts that have been taught.

2- Possibility to create student groups:

Student groups provide an excellent environment for teamwork, which increases opportunities for learning and participation among students.

3- Precise tracking of students' progress:

by using data analytics, we can track students' progress in each topic and thus identify strengths, weaknesses, and parts that represent obstacles that were not significantly clarified during the study.

4- Adding a recommendation system :

By using AI, we can track students' progress, thus we can create a recommendation system that recommends students to specific parts of the curriculum by studying their activity and statistics for each of them individually.

References

1. <https://arxiv.org/pdf/1803.08721.pdf>
2. https://huggingface.co/docs/transformers/model_doc/t5#overview
3. <https://huggingface.co/course/chapter1/1?fw=pt>
4. <https://nanonets.com/blog/ocr-with-tesseract/>
5. <https://pypi.org/project/pdf2image/>
6. <https://pypi.org/project/poppler-utils/>
7. <https://pillow.readthedocs.io/en/stable/>
8. <https://pypi.org/project/cleantext/>
9. <https://www.geeksforgeeks.org/python-opencv-cv2-imread-method/>
10. <https://docs.python.org/3/library/subprocess.html>
11. <https://towardsdatascience.com/sequencematcher-in-python-6b1e6f3915fc>
12. https://www.tutorialspoint.com/scikit_learn/scikit_learn_introduction.htm
13. <https://www.analyticsvidhya.com/blog/2021/11/how-sklearns-tfidfvectorizer-calculates-tf-idf-values/>
14. <https://www.learnpythonwithrune.org/performance-comparison-of-numba-vs-vectorization-vs-lambda-function-with-numpy/>
15. <https://www.geeksforgeeks.org/>
16. <https://flutter.dev/>
17. <https://dart.dev/>
18. <https://firebase.google.com/>
19. <https://www.postman.com/product/what-is-postman/>
20. <https://www.durgasoftonline.com/s/store/courses/description/Django-with-Real-Time-Project-Old-and-New-Videos#tableofcontents>
21. <https://www.durgasoftonline.com/s/store/courses/description/Django-REST-API-DRF>
22. <https://www.youtube.com/playlist?list=PL-osiE80TeTt2d9bfVyTiXJA-UTHn6WwU>
23. <https://docs.djangoproject.com/en/4.0/>
24. <https://www.djangoproject.com/>