

SS GYAAN SYNC HUB
EDUCATION MANAGEMENT SYSTEM

A Project

Submitted in partial fulfillment of the requirements for the degree of
Master of Computer Applications
For the Project Work [PCA20P03L]

Guided By
Megha Gupta (CETPA)



SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)
DELHI NCR CAMPUS, GHAZIABAD, (U.P.)

DEPARTMENT OF COMPUTER APPLICATIONS
(FACULTY OF SCIENCE & HUMANITIES)

SRM INSTITUTE OF SCIENCE & TECHNOLOGY
DELHI NCR CAMPUS MODINAGAR

(Deemed to be University U/S – 3 of UGC Act. 1956)(F. 9-9/98-U.3)

SUBMITTED TO:
DR. VISHU V.
(Associate Professor & Head)

SUBMITTED BY:
SAARANSH SHARMA
RA2232241030004

Year-2024

S R M INSTITUTE OF SCIENCE & TECHNOLOGY

DELHI NCR CAMPUS, MODINAGAR

(FACULTY OF SCIENCE & HUMANITIES)

DEPARTMENT OF COMPUTER APPLICATIONS

Register No.:

RA2232241030004

BONAFIDE CERTIFICATE

Certified to be the bonafide record of the work done by **SAARANSH SHARMA** of MCA 2nd Year / 4th Semester for the Award of **Master's Degree** in the **Department of Computer Applications** in **Project Work [PCA20P03L]** during the Academic Year 2023-24.

PROJECT – GUIDE

HEAD – COMPUTER APPLICATIONS

Submitted for the project viva held on _____.

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

I take this opportunity to express my deep sense of gratitude to all those who have contributed significantly by sharing their knowledge and experience in the completion of this project work.

My first word of gratitude is to ***Director Dr. S. Viswanathan, Dean(S&H) Dr.Navin Ahlawat HOD-MCA Vishu V ma'am and Project Coordinator Dr. Rajeev Kumar Sharma***, for providing this kind of opportunity and guidance throughout the project.

My heart full thanks to ***My Project Guide Dr. Vishu V.***, without her continuation help the project would not have been materialized in the present form. I am thankful to her for providing me with necessary insights and valuable suggestions helped me at every step. We would like to take this opportunity to extend our warm thoughts to those who helped me in making this project a wonderful experience.

Last but not least we would also like to thank our family and friends for their support and encouragement.

SAARANSH SHARMA

Reg. No.: RA2232241030004

Date: _____

DECLARATION OF STUDENT

I, **SAARANSH SHARMA**, a student of SRMIST, MODINAGAR, pursuing the Master of Computer Applications (MCA) program, hereby declare that the major project titled "EDUCATION MANAGEMENT SYSTEM" undertaken by me as part of the academic curriculum is designated as an industrial project.

Project Details:

- **Title: SS GYAN SYNC HUB: EDUCATION MANAGEMENT SYSTEM**
- **Duration: JAN 2024 to APRIL 2024**
- **Academic Year: 2023-2024**
- **Industrial Partner: CETPA INFOTECH PVT. LTD.**
- **Project Mentor (University): DR. VISHU V.**
- **Project Mentor (Industrial Partner): MEGHA GUPTA (CETPA)**

Declaration:

1. I declare that this major project is undertaken in collaboration with CETPA INFOTECH PVT. LTD., a recognized industry organization, as an integral part of my academic requirements for the MCA program.
2. The project work includes education platform for students to learn at the comfort, and it is aligned with the industry standards and practices.
3. I acknowledge and express gratitude to CETPA INFOTECH PVT. LTD., for providing the necessary resources, guidance, and support during the project duration.
4. I affirm that the project work is original, and any external sources or references used are appropriately cited in the project documentation.
5. I assure that the project work is conducted in compliance with ethical standards, and all applicable laws and regulations are adhered to.
6. I take full responsibility for the outcomes and results of the project presented in the project report and during the evaluation process.
7. I understand that the project report and related materials may be used for academic and promotional purposes by SRMIST, MODINAGAR and CETPA INFOTECH PVT. LTD.
8. I agree to abide by any confidentiality agreements or terms set forth by CETPA INFOTECH PVT. LTD., regarding proprietary information and sensitive data shared during the project.

Date: 25/04/2024

Place: _____

SS GYAAN SYNC HUB

EDUCATION MANAGEMENT SYSTEM





Ref No:- INT/CIPL/001/NOIDA

Date:- 8th April 2024

To Whomsoever It May Concern

This is to certify that Mr. Saaransh Sharma has worked as an "Intern in application development and done a project on Education Management System. He has successfully completed his internship with 75% of attendance.

His internship tenure was from 8th January 2024 to 8th April 2024.

During the span, we found him to be a punctual and hardworking person. He has completed all his assignments on time. His learning powers are good and he picks up swiftly. His feedback and evaluation proved that he learned keenly. Moreover, his interpersonal and communication skills are brilliant.



CETPA Infotech Pvt Ltd

HR Department

TABLE OF CONTENTS

1. CHAPTER 1: INTRODUCTION	1
1.1. PROJECT INTRODUCTION	1
1.2. PROBLEM DESCRIPTION	1
1.3. ABOUT ORGANIZATION	2
2. CHAPTER 2: PLATFORM USED	3
2.1. HARDWARE REQUIREMENTS	3
2.2. SOFTWARE REQUIREMENTS	3
2.3. FRONT & BACK END USED	4
3. CHAPTER 3: PROJECT ANALYSIS	6
3.1. SYSTEM STUDY	6
3.2. FEASIBILITY STUDIES	6
3.3. COST-BENEFIT ANALYSIS	7
4. CHAPTER 4: PROJECT DESIGN	8
4.1. SYSTEM ANALYSIS	8
4.1.1. SYSTEM FLOWCHARTS	8
4.2. SYSTEM DESIGN	10
4.2.1. FILE / DATABASE DESIGN	10
4.2.2. NORMALIZATION	11
4.2.3. SCREEN DESIGN	15
4.2.4. REPORT DESIGN	23
5. CHAPTER 5: TABLE STRUCTURE	24
6. CHAPTER 6: DIAGRAMS	28
6.1. E-R DIAGRAMS	28
6.2. DATA FLOW DIAGRAM (DFDS)	29
6.3. PERT CHART	30
6.4. USE CASE DIAGRAM	32
6.5. DATA DICTIONARY	33
7. CHAPTER 7: CODING	35
8. CHAPTER 8: FORM DESIGN	144
9. CHAPTER 9: TESTING MECHANISMS USED	176
9.1. EMULATOR TESTING	176
9.2. MANUAL TESTING	176
10. CHAPTER 10: IMPLEMENTATION MECHANISMS	177
11. CHAPTER 11: CONCLUSIONS	179
11.1. FINDINGS	179
11.2. LIMITATIONS	180
11.3. SCOPE FOR FUTURE PROSPECTS	180
12. CHAPTER 12: BIBLIOGRAPHY	181
APPENDICES	183
(a) SOFTWARE REQUIREMENT SPECIFICATIONS (SRS)	183
(b) USER MANUAL: INTERACTIVE LEARNING PLATFORM	186
PLAGIARISM CHECKER	188

CHAPTER 1: INTRODUCTION

CHAPTER 1: INTRODUCTION

1.1. PROJECT INTRODUCTION

In the world of current training, the panorama is always evolving, pushed with the aid of virtual innovation and technological improvements. Amidst this dynamic environment, the advent of SS GYAN SYNC HUB marks a sizeable milestone—an innovative Education Management System (EMS) designed to redefine how educators and newbies interact with educational content.

The task, SS GYAN SYNC HUB, embodies a commitment to harnessing technology to beautify the academic enjoy. It represents a fusion of modern-day hardware, state-of-the-art software program equipment, and statistics-pushed insights aimed toward empowering both educators and students alike. With its intuitive interface, robust functionality, and emphasis on connectivity, SS GYAN SYNC HUB seeks to go beyond conventional limitations and bring in brand-new technology for digital knowledge.

1.2. PROBLEM DESCRIPTION

Traditional education systems have long grappled with inefficiencies and obstacles, hindering the capacity for meaningful studying stories. Educators regularly face demanding situations in managing administrative responsibilities, handing over enticing content material, and successfully assessing student development. Similarly, college students stumble upon limitations inclusive of limited entry to academic sources, lack of personalized studying pathways, and problems in staying encouraged and related.

Moreover, the fast pace of technological advancement has highlighted the widening hole among conventional educational practices and the expectancies of the digital-native generation. With a growing reliance on mobile devices, cloud-based total answers, and interactive studying equipment, there may be a pressing want to modernize schooling systems and adapt to the evolving needs of novices and educators.

Recognizing these demanding situations, SS GYAN SYNC HUB emerges as a comprehensive solution—a platform designed to address the pain factors of traditional education control and free up new opportunities for collaboration, engagement, and customized studying.

1.3. ABOUT ORGANIZATION



CETPA INFOTECH PVT. LTD is a main pressure in specialized education, consultancy, and development services. With a wealthy legacy of excellence and innovation, CETPA INFOTECH has hooked up itself as a trusted partner in training and generation.

CETPA INFOTECH is a certified ISO 9001:2015 entity, upholding the highest standards of pleasantness throughout its various services. The corporation's dedication to excellence is pondered in its comprehensive education packages, job consultancy offerings, and impactful improvement tasks. Over the years, CETPA INFOTECH has empowered over 2 hundred,000 college students and experts, equipping them with the capabilities and knowledge needed to succeed in ultra-modern competitive panorama.

At the center of CETPA INFOTECH's philosophy is a willpower to practical learning, enterprise relevance, and consumer delight. Through its schooling projects, the employer bridges the gap between instructional principle and actual-international software, making ready rookies for the challenges of the contemporary body of workers. Additionally, CETPA INFOTECH's consultancy offerings facilitate professional possibilities for licensed students, leveraging its huge community of industry connections to healthy expertise with relevant activity opportunities.

In the area of software program and embedded development, CETPA INFOTECH stands as a beacon of innovation, delivering contemporary answers that drive business boom and achievement. With a crew of professional specialists and a consumer-centric technique, the company remains devoted to pushing the bounds of technology and delivering value to its customers.

CHAPTER 2:

PLATFORM USED

CHAPTER 2: PLATFORM USED

2.1. HARDWARE REQUIREMENTS

LAPTOP

Edition	Windows 10 Home Single Language
Processor	Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz 2.59 GHz
Installed RAM	8.00 GB (7.87 GB usable)
System type	64-bit operating system, x64-based processor

MOBILE

Processor	2.2Ghz MHz octa-core
RAM	4GB, 6GB, 8GB
Internal storage	64GB, 128GB (Expandable storage, microSD, 1TB)
Operating system	Android 13
Skin	One UI 5.0

2.2. SOFTWARE REQUIREMENTS



ANDROID STUDIO is a computer program made for making apps for Android phones. It's made by Google and it helps developers create and test their apps. (**FRONT END**)



FIREBASE is a tool made by Google for making apps for phones and websites. It helps developers create, improve, and expand their apps. (**BACK END**)

2.3. FRONT & BACK END USED

In the improvement of SS GYAN SYNC HUB, two key technologies play a pivotal function: Android Studio and Firebase. These effective tools, whilst included seamlessly, beautify the functionality, reliability, and scalability of the Education Management System (EMS), empowering both educators and college students with an immersive and interactive studying experience.

Android Studio:

- ***User Interface Design:*** Android Studio presents a complete set of tools for designing intuitive and visually attractive consumer interfaces. With its drag-and-drop format editor and XML-primarily based format files, builders can create responsive and consumer-friendly interfaces that beautify the general consumer enjoy of SS GYAN SYNC HUB.
- ***Application Logic:*** Java serves as the backbone of the utility logic in SS GYAN SYNC HUB, enabling developers to enforce middle functionalities including person authentication, information manipulation, and communication with the backend server. By leveraging Java's item-oriented ideas and huge libraries, developers can write smooth, maintainable code that meets the challenge's necessities.
- ***Compatibility and Optimization:*** Android Studio offers built-in equipment for checking out, debugging, and optimizing Android programs, making sure compatibility throughout different gadgets and versions of the Android running system. Through capabilities such as emulators, code profiling, and performance tracking, builders can pick out and address issues associated with performance, responsiveness, and aid utilization, resulting in a smoother and greater efficient consumer experience.

Firebase:

- ***Real-time Database:*** Firebase Realtime Database enables developers to keep and sync statistics in real-time between the Android utility and the backend server. This functionality is essential for SS GYAN SYNC HUB, making an allowance for seamless updates to course materials, student development, and communication channels. With Firebase's scalable and reliable infrastructure, SS GYAN SYNC HUB can accommodate a massive number of customers and make sure steady records get entry to and synchronization across devices.

- **Authentication:** Firebase Authentication gives sturdy person authentication and authorization mechanisms, allowing customers to soundly register to SS GYAN SYNC HUB the usage of their electronic mail addresses, social media money owed, or telephone numbers. By integrating Firebase Authentication, builders can put into effect steady access controls, shield touchy person facts, and ensure a continuing login enjoy for educators and students.
- **Cloud Storage:** Firebase Cloud Storage offers scalable and stable cloud garage solutions for storing multimedia content material such as PDFs, videos, and snap shots. This characteristic is critical for SS GYAN SYNC HUB, permitting educators to upload and percentage instructional assets with college students seamlessly. With Firebase Cloud Storage, SS GYAN SYNC HUB can accommodate large files, ensure information integrity, and offer fast and reliable get entry to to educational materials, improving the general learning revel in.

CHAPTER 3:

PROJECT ANALYSIS

CHAPTER 3: PROJECT ANALYSIS

A thorough assignment evaluation became carried out to evaluate the present educational structures, pick out their obstacles, and endorse a solution that addresses the wishes and demanding situations faced via educators and students. This analysis encompassed a complete gadget have a look at, feasibility studies, and value-benefit analysis to make sure the viability and effectiveness of the proposed Education Management System (EMS).

3.1. SYSTEM STUDY

The device takes a look at involved an in-depth examination of the prevailing instructional systems, such as their functionalities, strategies, and shortcomings. It turned into found that conventional training systems often struggled with inefficiencies in administrative duties, boundaries in having access to educational sources, and the dearth of customized studying pathways for college kids. These challenges hindered the overall effectiveness and engagement within the gaining knowledge of technique.

In reaction to these shortcomings, the proposed SS GYAN SYNC HUB turned into designed with particular targets in mind:

- Streamlining administrative responsibilities for educators
- Providing seamless get right of entry to educational resources for college students
- Offering personalized mastering pathways to cater to person scholar wishes
- Enhancing connectivity and engagement inside the learning manner

3.2. FEASIBILITY STUDIES

- ***Economic Feasibility:*** The economic feasibility has a look at analyzed the financial implications of developing and keeping SS GYAN SYNC HUB. It considered elements inclusive of development prices, operational prices, and capacity returns on funding. Despite initial funding requirements, the lengthy-time period advantages of enhancing instructional effects and performance justified the mission's financial feasibility.
- ***Technical Feasibility:*** The technical feasibility examines examined the technological necessities and challenges related to imposing SS GYAN SYNC HUB. It evaluated the

availability of necessary hardware and software program assets, compatibility with current infrastructure, and the knowledge required for improvement and maintenance. With improvements in cellular era, cloud computing, and software program improvement tools, SS GYAN SYNC HUB changed into deemed technically possible to put in force.

- ***Duration:*** The feasibility research additionally assessed the expected duration for developing and deploying SS GYAN SYNC HUB. This involved studying the complexity of the project, useful resource availability, and capacity dangers and challenges. While the improvement timeline became dependent on various factors, together with challenge scope and resource allocation, careful planning and task control techniques have been employed to make sure timely delivery.

3.3. COST-BENEFIT ANALYSIS

A cost-benefit evaluation became conducted to weigh the capacity expenses of imposing SS GYAN SYNC HUB in opposition to the anticipated blessings and returns on funding. This analysis considered each tangible and intangible charges and benefits related to the project, which include improvement charges, operational expenses, stepped forward educational effects, and stronger consumer delight.

The findings of the price-gain analysis indicated that the advantages of SS GYAN SYNC HUB, including increased performance in academic management, stepped forward scholar engagement and performance, and improved accessibility to instructional assets, outweighed the preliminary investment and ongoing operational fees. As such, the mission become deemed economically possible and beneficial for all stakeholders concerned—educators, students, and the corporation.

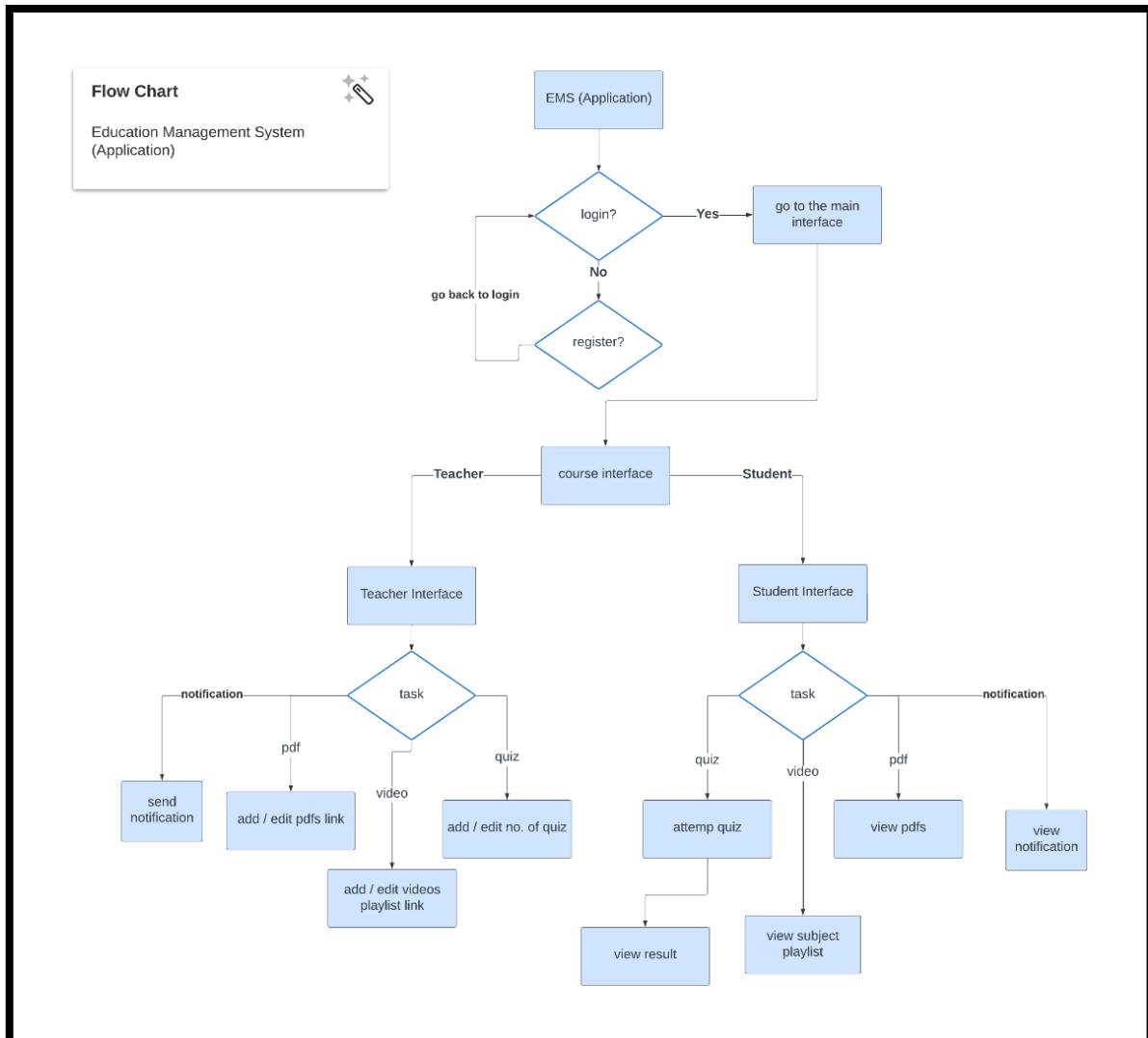
CHAPTER 4:

PROJECT DESIGN

CHAPTER 4: PROJECT DESIGN

4.1. SYSTEM ANALYSIS

4.1.1. System Flowcharts



Steps:

1. Login Process:

- a. Enter credentials (username and password).
- b. Check if the user is registered and credentials are correct.
- c. If login is successful, proceed to step 3.
- d. If not logged in, proceed to step 2.

2. Registration Process:

- a. Provide necessary information (name, email, password, etc.).
- b. Choose whether to register as a teacher or student.
- c. Submit the registration form.
- d. Redirect back to the login page.

3. Main Interface:

- a. Upon successful login, redirect to the main interface.
- b. Display different courses available.

4. Actions for Teacher:

a. If logged in as a teacher:

- i. Display options to add notifications, PDF links, video playlist links, and quizzes.
- ii. Allow the teacher to add/edit notifications.
- iii. Allow the teacher to add/edit pdf links.
- iv. Allow the teacher to add/edit video playlist links.
- v. Allow the teacher to add/edit quizzes.

5. Actions for Student:

a. If logged in as a student:

- i. Display notifications.
- ii. Display PDF links.
- iii. Display video playlist links.
- iv. Allow students to attempt quizzes and receive results.

6. Logout Option:

- a. Provide a logout option for both teachers and students.
- b. Upon logout, redirect back to the login page.

4.2. SYSTEM DESIGN

4.2.1. File / Database Design

education management system ▾

Realtime Database

Data Rules Backups Usage  Extensions

 <https://education-management-sys-e2d6e-default-rtdb.firebaseio.com>

<https://education-management-sys-e2d6e-default-rtdb.firebaseio.com/>



4.2.2. Normalization

To normalize the given database into different normal forms, we'll first identify the functional dependencies and then create tables based on those dependencies.

Functional Dependencies:

- First Normal Form (1NF):
 - All attributes must have atomic values.
 - There should be no repeating groups or arrays.
- Second Normal Form (2NF):
 - Must be in 1NF.
 - All non-key attributes are fully functionally dependent on the primary key.
- Third Normal Form (3NF):
 - Must be in 2NF.
 - No transitive dependencies.
- Boyce-Codd Normal Form (BCNF):
 - Must be in 3NF.
 - Every determinant must be a candidate key.
- Fourth Normal Form (4NF):
 - Must be in BCNF.
 - No multi-valued dependencies.
- Fifth Normal Form (5NF):
 - Must be in 4NF.
 - No join dependencies.

Tables: (Based on the given JSON records, here are the normalized tables)

Users Table:

user_id	username	Email	name	password	phone
0	meghaG	meghaG@gmail.com	Megha Gupta	****	****
1	ssharma04	ssharma04@gmail.com	Saaransh Sharma	****	****

Course Table:

course_id	course_name	course_icon
1	MCA	[Link]
2	BTECH	[Link]
3	BCA	[Link]

Subject Table:

subject_id	subject_name	subject_icon	course_id (FK)
11	Java	[Link]	1
12	C/C++	[Link]	1
13	Python	[Link]	1
14	Android	[Link]	1
15	Web	[Link]	1
11	Java	[Link]	2
12	C/C++	[Link]	2
13	Python	[Link]	2
14	Android	[Link]	2
15	Web	[Link]	2
11	Java	[Link]	3
12	C/C++	[Link]	3
13	Python	[Link]	3
14	Android	[Link]	3
15	Web	[Link]	3

PDF Table:

subject_id	pdfs_link
(FK)	
11	[Link]
12	[Link]
13	[Link]
14	[Link]
15	[Link]

Video Table:

subject_id	videos_link
(FK)	
11	[Link]
11	[Link]
12	[Link]
12	[Link]
13	[Link]
13	[Link]
14	[Link]
14	[Link]
15	[Link]
15	[Link]

MCQ Table:

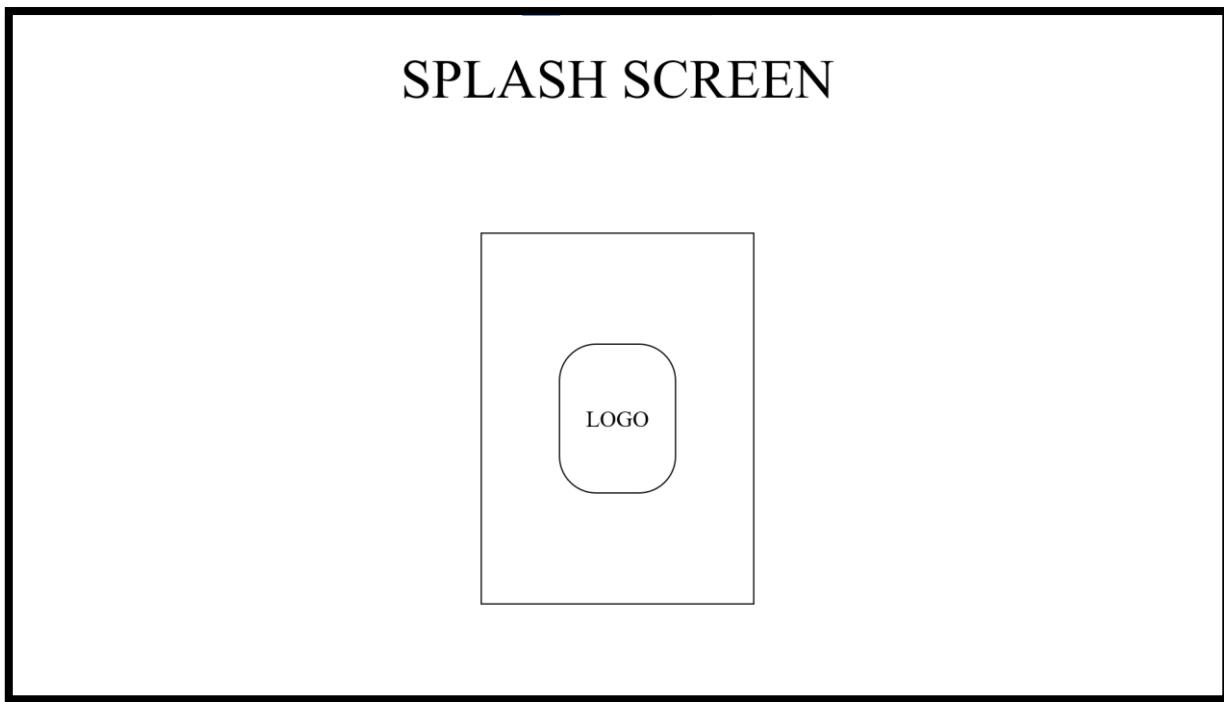
subject_id (FK)	Question	option_1	option_2	option_3	option_4	correct
11	What is the main purpose of the "System.out.println()" statement in Java?	a) Reading user input	b) Declaring variables	c) Printing output to the console	d) Performing arithmetic operations	3
...

Notification Table:

notification_id	Title	message	course_id	date_time
			(FK)	
Fri Mar 29 12:13:18 GMT+05:30 2024	JAVA	Attention Students! Exciting news! A new test has been uploaded for your review. Please log in to your accounts to access it. Good luck with your preparation!	1	Fri Mar 29 12:13:18 GMT+05:30 2024
Fri Mar 29 12:13:41 GMT+05:30 2024	For All Subjects	Heads up, students! A new PDF has just been uploaded for you all. Make sure to check it out on your accounts for the latest information. Happy reading!	2	Fri Mar 29 12:13:41 GMT+05:30 2024
...

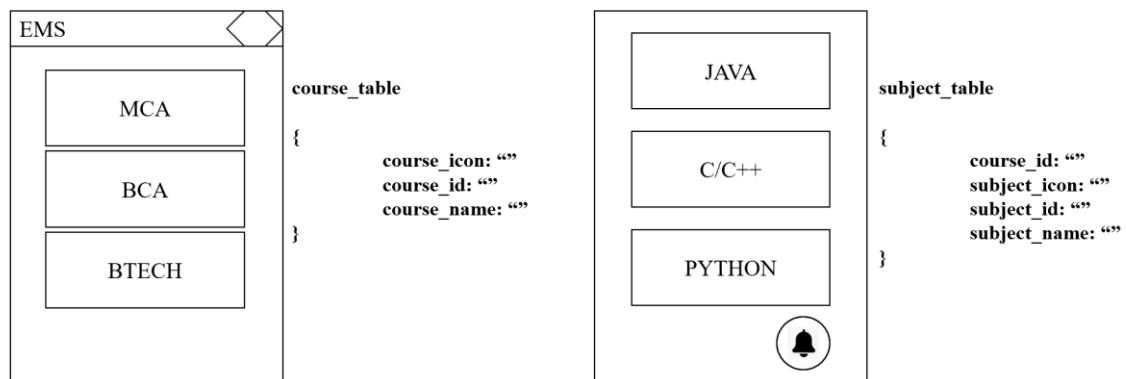
These tables are normalized as much as 3NF as in step with the furnished statistics. Further normalization to BCNF, 4NF, and 5NF may additionally require extra evaluation of functional dependencies.

4.2.3. Screen Design

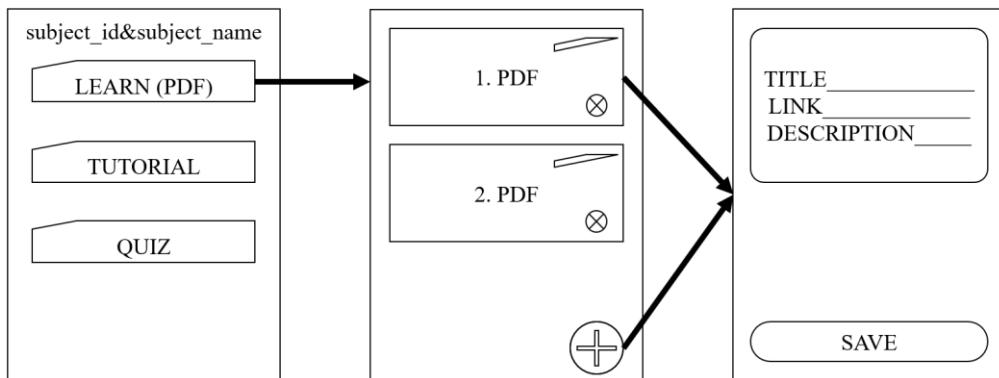


SIGN IN FORM	REGISTRATION FORM
<p>Email _____</p> <p>Password _____</p> <p><input type="button" value="Login"/></p> <p>Don't have an account? Register here</p>	<p>SHARED PREFERENCE</p> <p>{ DATA }</p> <p>Already register: login to the Course Pannel.</p> <p>If not Register: create an account.</p> <p><input type="radio"/> Teacher <input type="radio"/> Student</p> <p><input type="button" value="Sign In"/></p> <p>Already have an account? Login here</p> <pre>registration_form_table { email: "" name: "" phone: "" password: "" id: 1 (teacher), 2 (student) }</pre>

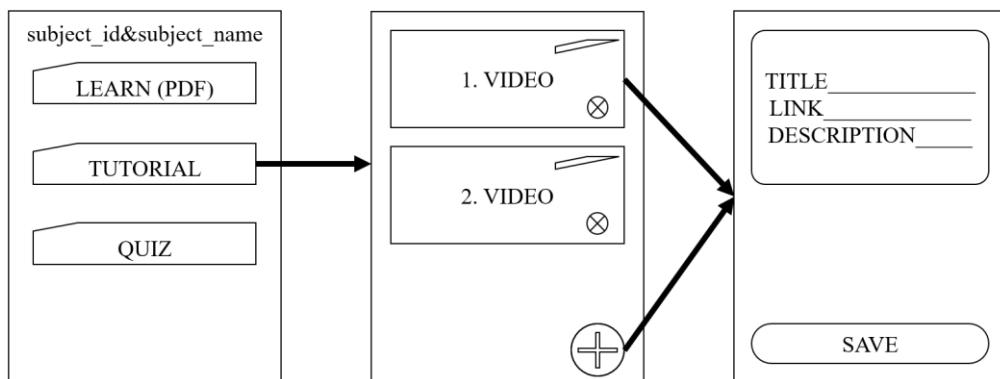
COURSE (TEACHER) SUBJECT (TEACHER)



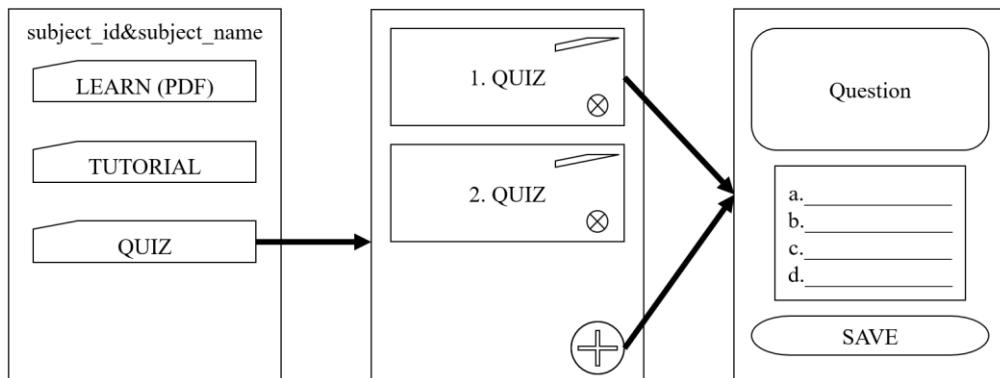
TEACHER



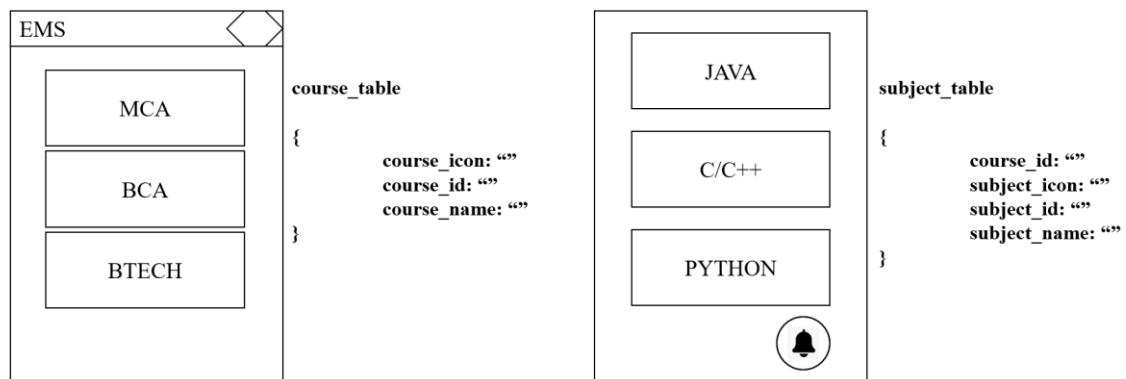
TEACHER



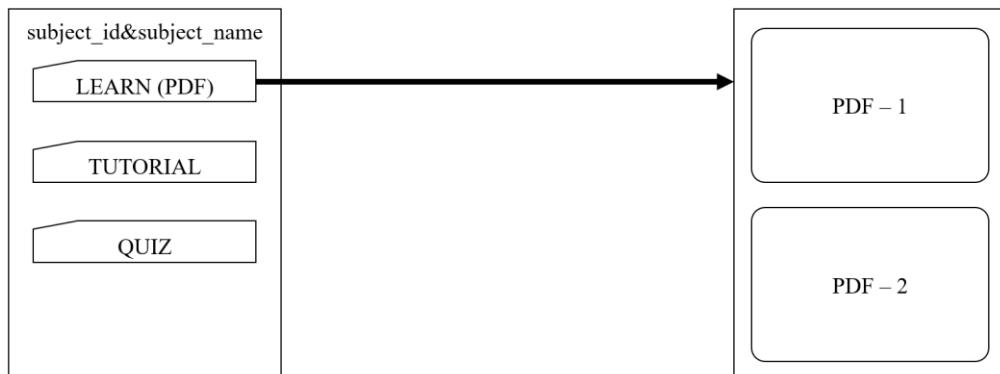
TEACHER



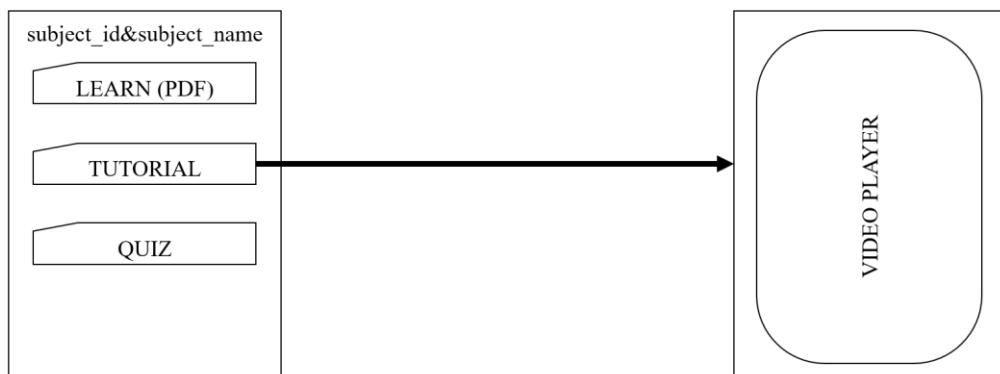
COURSE (STUDENT) SUBJECT (STUDENT)



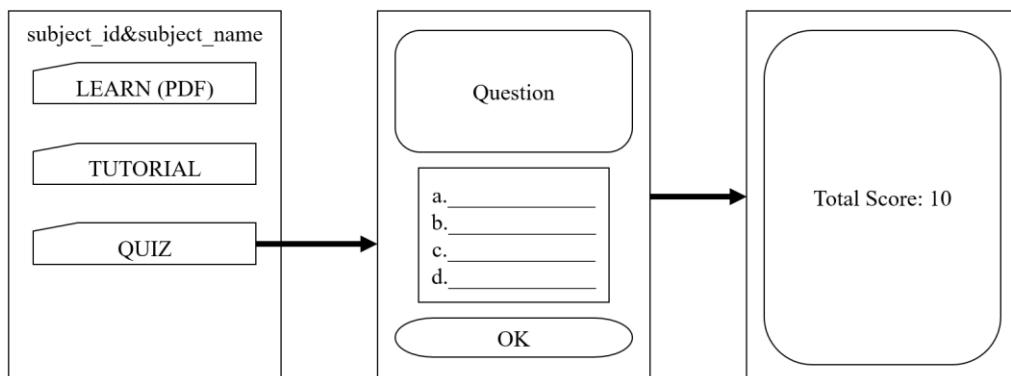
STUDENT



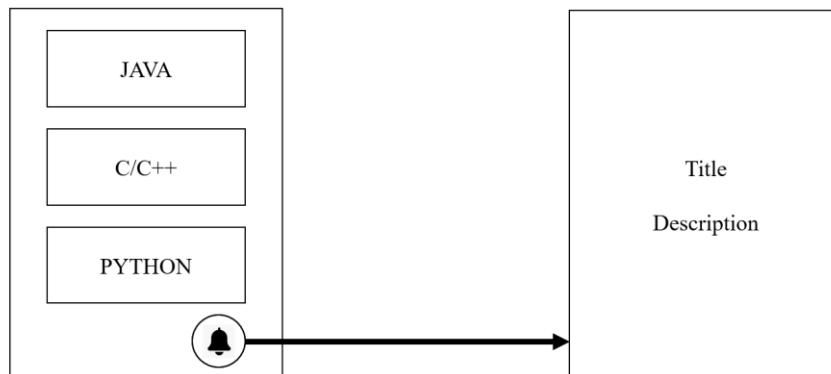
STUDENT



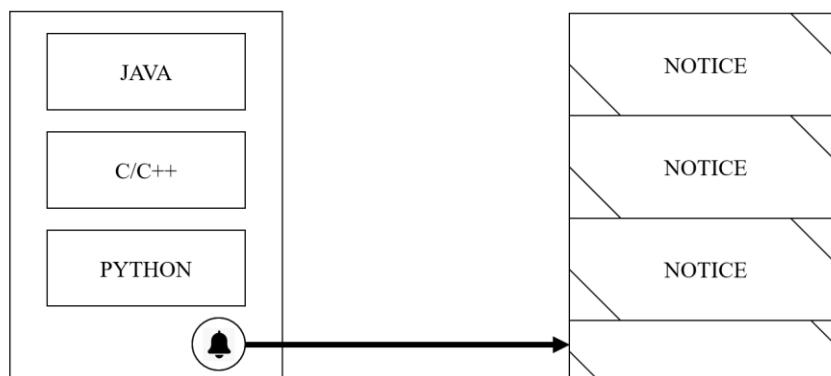
STUDENT



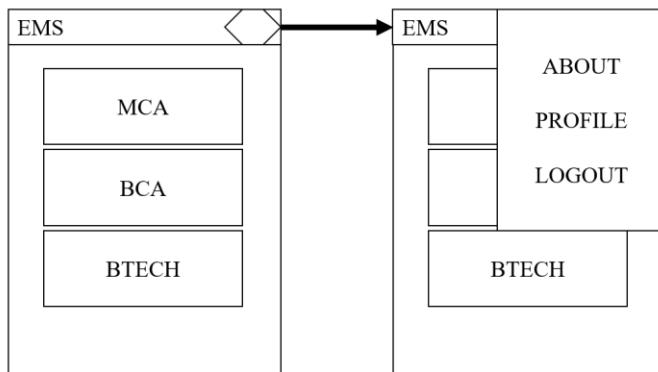
NOTIFICATION SCREEN (TEACHER)



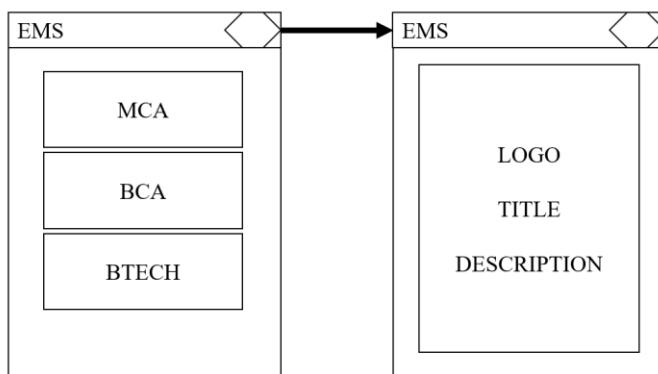
NOTIFICATION SCREEN (STUDENT)



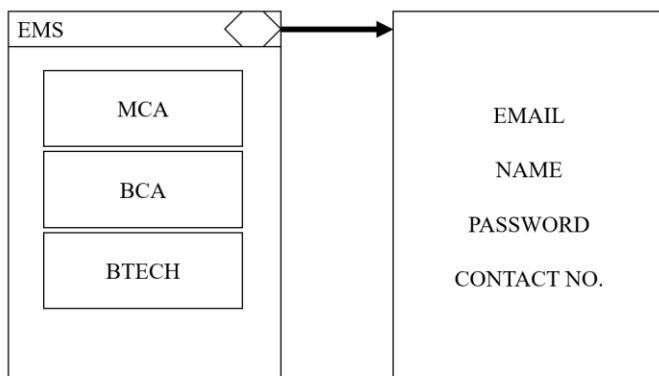
MENU SCREEN



ABOUT SCREEN



PROFILE SCREEN



4.2.4. Report Design

The file delves into the complete evaluation of person revel in (UX) inside an internet utility devoted to direction control. It meticulously examines key degrees of user interplay, including the login and registration processes, navigation via the primary interface, and the utilization of functions with the aid of both teachers and college students.

The login procedure, which serves because the gateway to the platform, is scrutinized for its efficiency, effectiveness, and usual person satisfaction. Similarly, the registration technique is evaluated in terms of completeness, clarity, and the seamless transition it offers customers from registration to logging in.

The important interface navigation is classified based totally on standards inclusive of accessibility, readability, and engagement, reflecting on the convenience with which users can discover and interact with diverse courses. Moreover, the document analyzes the capabilities to be had to teachers, such as notifications, PDF hyperlinks, video playlists, and quizzes, with a focal point on functionality, usability, and customization options. Likewise, the features on hand to college students, inclusive of notifications, PDFs, video playlists, and quizzes, are evaluated in terms of accessibility, interactivity, and readability of remarks supplied on quiz performance.

Furthermore, the record addresses the logout process, emphasizing the importance of visibility and consumer delight on this very last stage of interaction.

Lastly, the report offers a holistic evaluation of the UX within the net utility, presenting insights into areas of strength and possibilities for improvement. Recommendations derived from the assessment intention to streamline approaches, beautify accessibility, improve function functionality, and optimize the overall person experience. Additionally, the appendix might also encompass supplementary statistics, charts, or person remarks amassed in the course of the assessment technique for further reference and evaluation.

CHAPTER 5:

TABLE STRUCTURE

CHAPTER 5: TABLE STRUCTURE

USERS					
USERNAME	EMAIL	ID	NAME	PASSWORD	PHONE
meghaG	meghaG@gmail.com	0	Megha Gupta	*****	*****
ssharma04	ssharma04@gmail.com	1	Saaransh Sharma	*****	*****
COURSE					
COURSE_ID	COURSE_NAME	COURSE_ICON			
1	MCA	Link			
2	BTECH	Link			
3	BCA	Link			
SUBJECT					
SUBJECT_ID	SUBJECT_NAME	COURSE_ID	SUBJECT_ICON		
11	Java	1	Java Icon		
12	C/C++	1	C/C++ Icon		
13	Python	1	Python Icon		
14	Android	1	Android Icon		
15	Web	1	Web Icon		
11	Java	2	Java Icon		
12	C/C++	2	C/C++ Icon		
13	Python	2	Python Icon		
14	Android	2	Android Icon		
15	Web	2	Web Icon		
11	Java	3	Java Icon		
12	C/C++	3	C/C++ Icon		
13	Python	3	Python Icon		
14	Android	3	Android Icon		
15	Web	3	Web Icon		

DATA_PDF

SUBJECT_ID	PDFS_LINK
11	Java Programming PDF
12	OOPS Lecture Notes PDF
13	Python PDF
14	C++ Sample Pages PDF
15	HTML, CSS, JS Combined PDF

DATA_VIDEO

SUBJECT_ID	VIDEOS_LINK
11	Java Playlist 1
11	Java Playlist 2
12	C++ Playlist 1
12	C++ Playlist 2
13	Python Playlist 1
13	Python Playlist 2
14	Android Playlist 1
14	Android Playlist 2
15	Web Playlist 1
15	Web Playlist 2

NOTIFICATION

NOTIFICATION_ID	COURSE_ID	TITLE	DATE_TIME	MESSAGE
Fri Mar 29 12:13:18 GMT+05:30 2024	1	JAVA	Fri Mar 29 12:13:18 GMT+05:30 2024	Attention Students! Exciting news! A new test has been uploaded for your review. Please log in to your accounts to access it. Good

luck with your preparation!

Fri Mar 29 12:13:41 2 FOR ALL Fri Mar 29 Heads up,
GMT+05:30 2024 SUBJECTS 12:13:41 students! A new
PDF has just been
uploaded for you
all. Make sure to
check it out on
your accounts for
the latest
information.
Happy reading!

Fri Mar 29 12:14:16 3 ANDROID Fri Mar 29 Attention
GMT+05:30 2024 NEW Students! Exciting
VIDEO news! A new video
has just been
uploaded for you
all. Head over to
your accounts to
watch and learn.
Don't miss out!

MCQ					
JAVA					
QUESTION	OPTION_1	OPTION_2	OPTION_3	OPTION_4	CORRECT
What is the main purpose of the "System.out.println()" statement in Java?	a) Reading user input	b) Declaring variables	c) Printing output to the console	d) Performing arithmetic operations	c) Printing output to the console
...					

C/C++

QUESTION	OPTION_1	OPTION_2	OPTION_3	OPTION_4	CORRECT
What is the purpose of the "printf()" function in C?	a) Reading input from the user	b) Declaring variables	c) Printing output to the console	d) Performing arithmetic operations	c) Printing output to the console

...

PYTHON

QUESTION	OPTION_1	OPTION_2	OPTION_3	OPTION_4	CORRECT
How is user input accepted in Python?	a) input()	b) read()	c)	d)	a) input() user_input()

...

ANDROID

QUESTION	OPTION_1	OPTION_2	OPTION_3	OPTION_4	CORRECT
What is Android Studio?	a) An Android device	b) An Android programming language	c) An Integrated Development Environment	d) A type of Android application language (IDE) for Android	c) An Integrated Development Environment (IDE) for Android

...

WEB

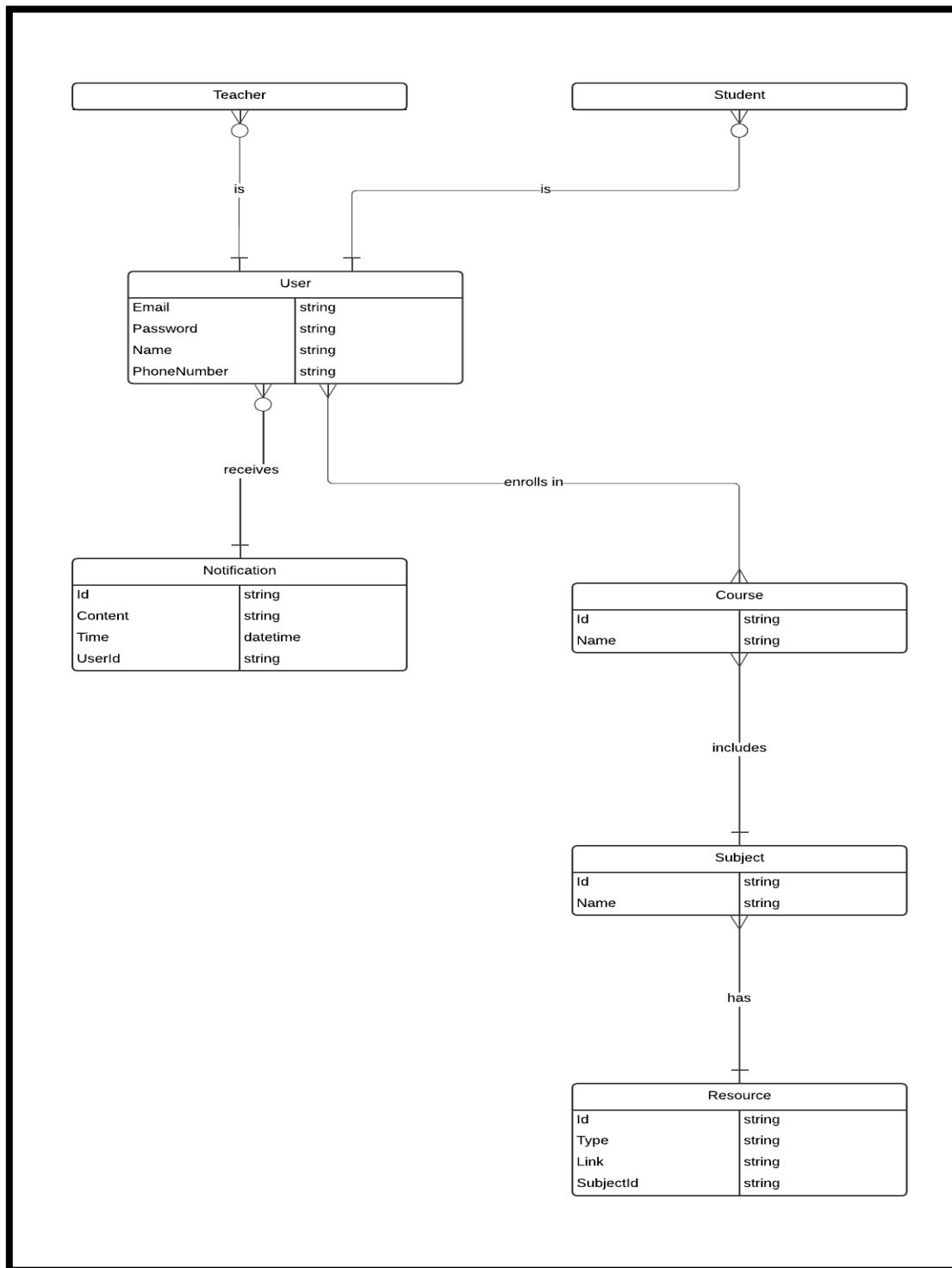
QUESTION	OPTION_1	OPTION_2	OPTION_3	OPTION_4	CORRECT
What does the term "WWW" stand for in the context of web development?	a) World Web Wide	b) Web World Wide	c) World Wide Web	d) Wide Web World	c) World Wide Web

...

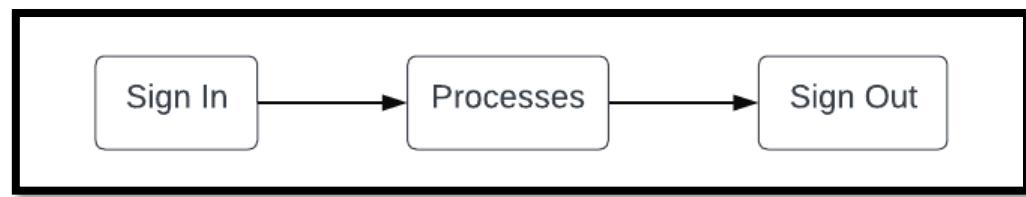
CHAPTER 6: DIAGRAMS

CHAPTER 6: DIAGRAMS

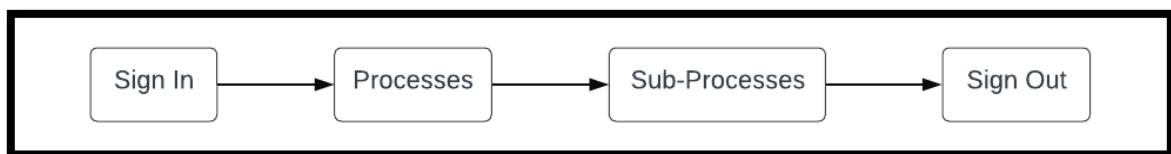
6.1. E-R Diagrams



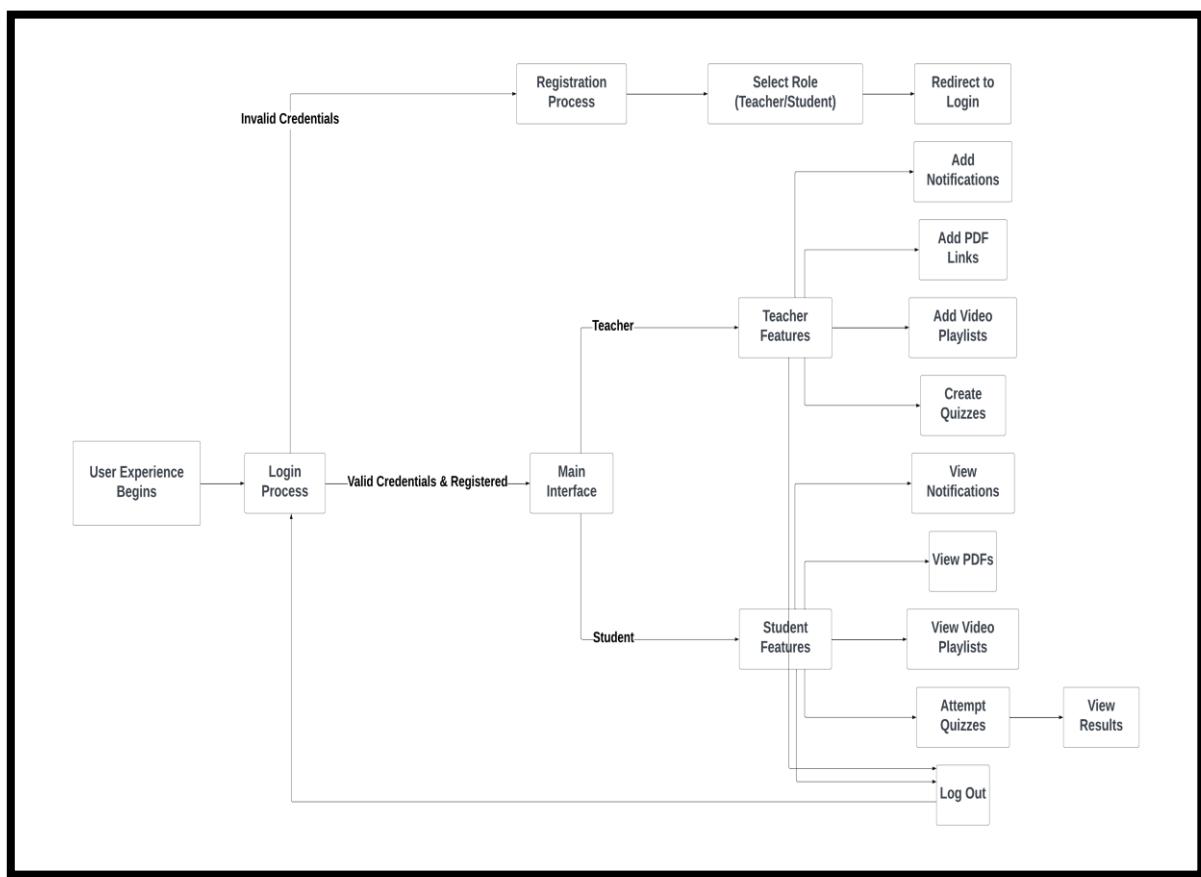
6.2. Data Flow Diagram (DFDs)



Data Flow Diagram (Level – 0)

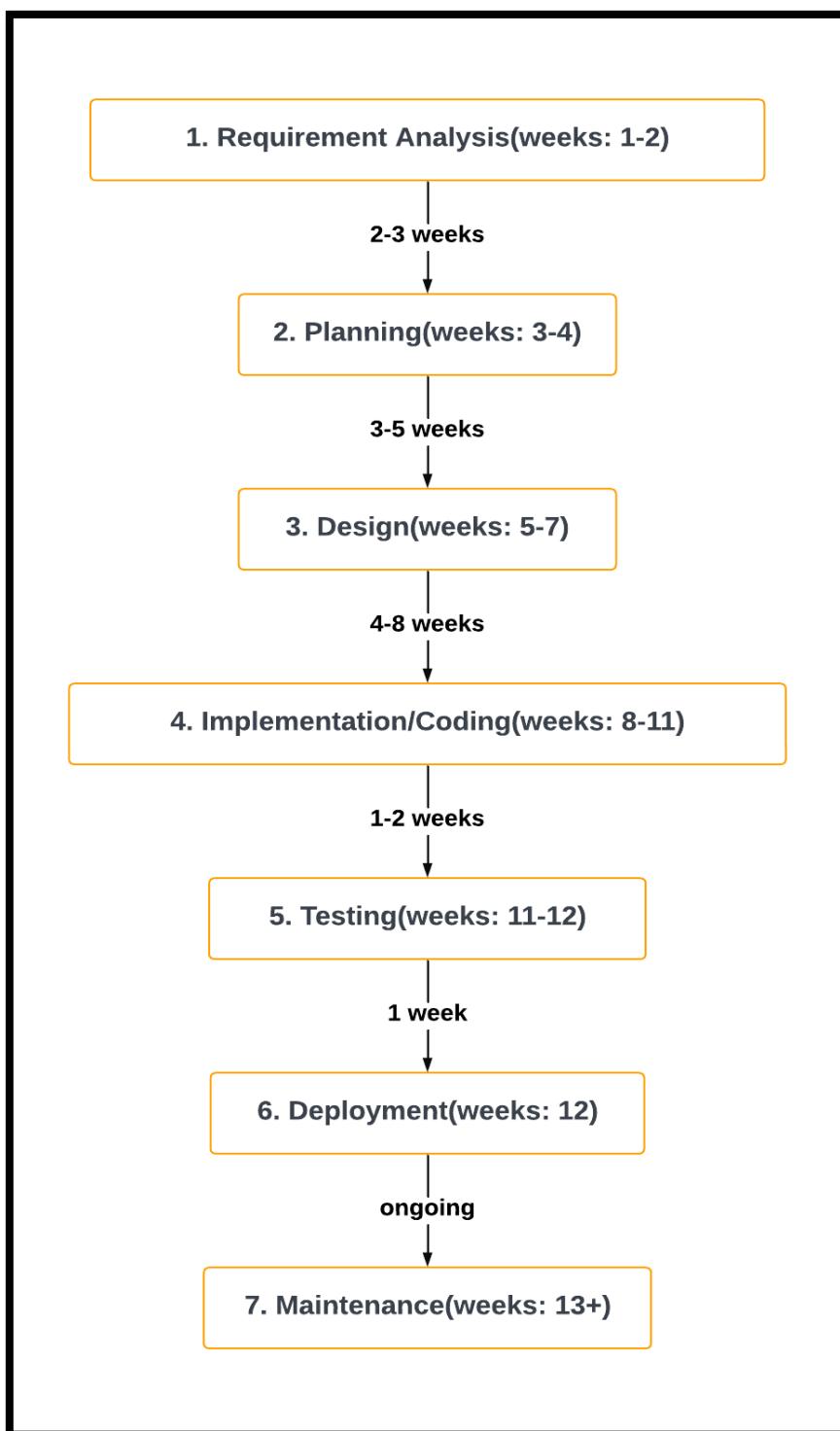


Data Flow Diagram (Level – 1)



Data Flow Diagram (Level – 2)

6.3. PERT Chart



TASK	Requirement Analysis	Planning	Design	Implementation /Coding	Testing	Deployment	Maintenance
WEEK 1	#####						
WEEK 2	#####						
WEEK 3		#####					
WEEK 4		#####					
WEEK 5		#####	#####				
WEEK 6			#####				
WEEK 7			#####				
WEEK 8				#####			
WEEK 9				#####			
WEEK 10				#####	#####		
WEEK 11					#####	#####	
WEEK 12						#####	#####
WEEK 13+						#####	#####

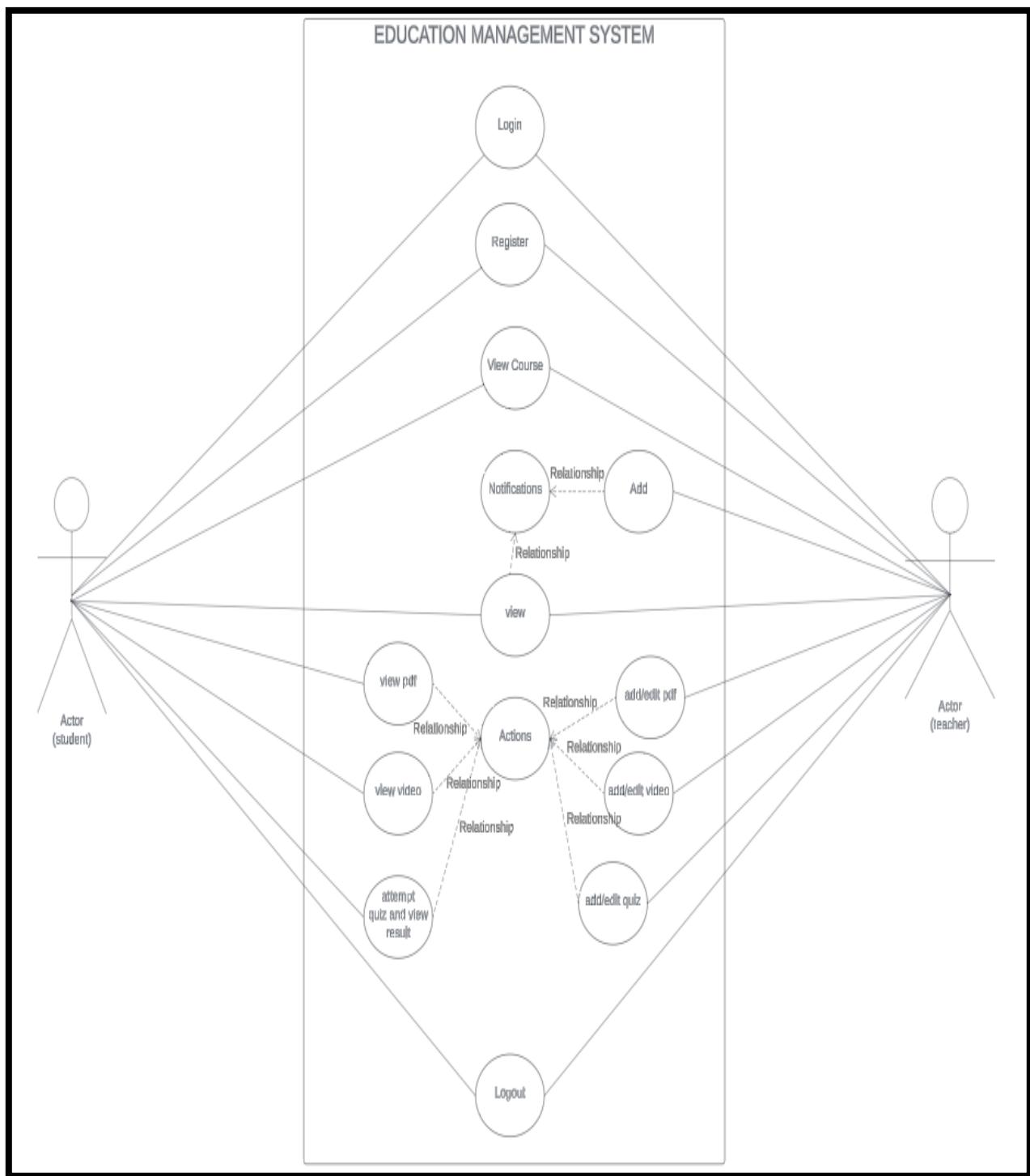
In this Gantt chart:

- Each row represents a venture in the task.
- Weeks are represented vertically.
- '#' symbols imply the period of every venture.
- Tasks that may run simultaneously are scheduled in parallel.
- The vital route, that is the longest course through the project and determines the shortest possible length for the venture, is highlighted by using the longest collection of connected responsibilities with none slack or flow.

In this example, the important path is:

Requirement Analysis -> Planning -> Design -> Implementation/Coding -> Testing -> Deployment -> Maintenance.

6.4. Use Case Diagram



6.5. Data Dictionary

User		
Field Name	Data Type	Description
email	String	Email address of the user
name	String	Name of the user
phone	String	Phone number of the user
password	String	Password of the user
id	Integer	Unique identifier for the user
...		
Course		
Field Name	Data Type	Description
courseCode	String	Code representing the course
courseTitle	String	Title or name of the course
course_icon	String	File path or URL to the icon representing course
...		
Subject		
Field Name	Data Type	Description
subjectId	String	Identifier for the subject
subjectName	String	Name of the subject
courseId	String	Identifier for the course to which the subject belongs
subject_icon	String	File path or URL to the icon representing the subject
...		
Action		
Field Name	Data Type	Description
actionName	String	Name or description of the action
actionId	Integer	Unique identifier for the action
...		

PDF

Field Name	Data Type	Description
subject_id	Integer	Identifier for the subject related to the PDF
pdfs_link	String	Link or path to the PDF file
...		

Video

Field Name	Data Type	Description
subject_id	String	Identifier for the subject related to the video
videos_link	String	Link or path to the video
...		

MCQ

Field Name	Data Type	Description
subject_id	Integer	Identifier for the subject of the MCQ
question	String	The actual question presented to the user
option_1	String	First option for the multiple-choice question
option_2	String	Second option for the multiple-choice question
option_3	String	Third option for the multiple-choice question
option_4	String	Fourth option for the multiple-choice question
correct	String	The correct option among the four options
...		

Notification

Field Name	Data Type	Description
notification_id	String	Identifier for the notification
title	String	Title of the notification
message	String	Body or content of the notification
course_id	String	Identifier for the associated course
date_time	String	Date and time when the notification was created
...		

CHAPTER 7:

CODING

CHAPTER 7: CODING

ACTIVITY_LOGO.XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical">

    <ImageView
        android:id="@+id/logoImageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="20dp"
        android:layout_marginBottom="20dp"
        android:src="@drawable/sss" />

</LinearLayout>
```

LOGOACTIVITY.JAVA

```
package com.example.newproject_testing;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import java.util.Timer;
import java.util.TimerTask;
public class logoActivity extends AppCompatActivity {

    MySharedPreference preference;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_logo);
        preference = new MySharedPreference(this);
        Timer timer = new Timer();
        timer.schedule(new TimerTask() {
            @Override
            public void run() {
                if(preference.isIs_login()) {
                    Intent i = new Intent(logoActivity.this, CourseActivity.class);
                    startActivity(i);
                    finish();
                }
            }
        else {
            Intent i = new Intent(logoActivity.this, loginActivity.class);
            startActivity(i);
            finish();
        }
    },2000);
    } }
```

ACTIVITY_LOGIN.XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="80dp">

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="314dp"
        android:layout_marginBottom="16dp"
        android:scaleType="centerCrop"
        android:src="@drawable/sss" />

    <EditText
        android:id="@+id/emailEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Email"
        android:inputType="textEmailAddress"
        android:layout_marginBottom="8dp"/>

    <EditText
        android:id="@+id/passwordEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Password"
        android:inputType="textPassword"
        android:layout_marginBottom="16dp"/>
```

```
<Button  
    android:id="@+id/loginButton"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Login"  
    android:layout_marginBottom="16dp"/>  
  
<TextView  
    android:id="@+id/registerTextView"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Register if you don't have an account"  
    android:textColor="@color/colorPrimary"  
    android:layout_gravity="center"/>  
  
</LinearLayout>
```

LOGINACTIVITY.JAVA

```
package com.example.newproject_testing;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import com.example.newproject_testing.models.mcq;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebaseio.database.DataSnapshot;
import com.google.firebaseio.database.DatabaseError;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
import com.google.firebaseio.database.ValueEventListener;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

public class loginActivity extends AppCompatActivity {
    EditText emailEditText, passwordEditText;
    Button loginButton;
    TextView registerTextView;
    MySharedPreference preference;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_login);
emailEditText = findViewById(R.id.emailEditText);
passwordEditText = findViewById(R.id.passwordEditText);
loginButton = findViewById(R.id.loginButton);
registerTextView = findViewById(R.id.registerTextView);
preference = new MySharedPreference(this);
loginButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        FirebaseDatabase database = FirebaseDatabase.getInstance();
        DatabaseReference myRef = database.getReference("users");
        myRef.child(emailEditText.getText().toString().split("@")[0]).get().addOnCompleteListener(new OnCompleteListener<DataSnapshot>() {
            @Override
            public void onComplete(@NonNull Task<DataSnapshot> task) {
                if (!task.isSuccessful()) {
                    Log.e("firebaseDATA", "Error getting data", task.getException());
                } else {
                    Log.e("firebaseDATA", String.valueOf(task.getResult().getValue()));
                    HashMap<String , Object> map = (HashMap<String, Object>) task.getResult().getValue();
                    String password = map.get("password").toString();
                    String emails = map.get("email").toString();
                    int type = Integer.parseInt(map.get("id").toString());
                    Log.e("firebaseDATA", password);
                    if(password.equals(passwordEditText.getText().toString())){
                        preference.setIs_login(true);
                        preference.setType_id(type);
                        preference.setEmail(emails);
                        Intent i = new Intent(loginActivity.this,CourseActivity.class);
                        startActivity(i);
                    }
                }
            }
        });
    }
});

```

```
        finish();
    }
    else {
        Toast.makeText(getApplicationContext(),"Password Not
Correct",Toast.LENGTH_LONG).show();
    }
}
});
});
registerTextView.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent i = new Intent(loginActivity.this,registrationActivity.class);
        startActivity(i);
    }
});
}
```

ACTIVITY_REGISTRATION.XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="50dp">

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="299dp"
        android:layout_marginBottom="16dp"
        android:scaleType="centerCrop"
        android:src="@drawable/sss" />

    <EditText
        android:id="@+id/emailEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Email"
        android:inputType="textEmailAddress"
        android:layout_marginBottom="8dp"/>

    <EditText
        android:id="@+id/nameEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Name"
        android:inputType="text"
        android:layout_marginBottom="8dp"/>
```

```
<EditText  
    android:id="@+id/phoneEditText"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:hint="Phone"  
    android:inputType="phone"  
    android:layout_marginBottom="8dp"/>
```

```
<EditText  
    android:id="@+id/passwordEditText"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:hint="Password"  
    android:inputType="textPassword"  
    android:layout_marginBottom="8dp"/>
```

```
<RadioGroup  
    android:id="@+id/groupByRadioGroup"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:orientation="horizontal"  
    android:layout_marginBottom="8dp">
```

```
<RadioButton  
    android:id="@+id/teacherRadioButton"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Teacher"/>
```

```
<RadioButton  
    android:id="@+id/studentRadioButton"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"
```

```
        android:text="Student"
        android:layout_marginStart="32dp"/>
    </RadioGroup>

    <Button
        android:id="@+id/registerButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Register"
        android:layout_marginBottom="16dp"/>

    <TextView
        android:id="@+id/alreadyRegisteredTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Already registered? Login"
        android:textColor="@color/colorPrimary"
        android:layout_gravity="center"/>

</LinearLayout>
```

REGISTRATIONACTIVITY.JAVA

```
package com.example.newproject_testing;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;
import android.widget.Toast;
import com.example.newproject_testing.models.user;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.pdftron.pdf.annots.RadioButtonGroup;

public class registrationActivity extends AppCompatActivity {
    EditText emailEditText, nameEditText , phoneEditText ,passwordEditText;
    RadioGroup groupByRadioGroup;
    RadioButton teacherRadioButton, studentRadioButton;
    Button registerButton;
    TextView alreadyRegisteredTextView;
    user userModel;
    int type_id;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_registration);
        emailEditText = findViewById(R.id.emailEditText);
        nameEditText = findViewById(R.id.nameEditText);
```

```

phoneEditText = findViewById(R.id.phoneEditText);
passwordEditText = findViewById(R.id.passwordEditText);
groupByRadioGroup = findViewById(R.id.groupByRadioGroup);
teacherRadioButton = findViewById(R.id.teacherRadioButton);
studentRadioButton = findViewById(R.id.studentRadioButton);
registerButton = findViewById(R.id.registerButton);
alreadyRegisteredTextView = findViewById(R.id.alreadyRegisteredTextView);
alreadyRegisteredTextView.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent i = new Intent(registrationActivity.this,loginActivity.class);
        startActivity(i);
    }
});
groupByRadioGroup.setOnCheckedChangeListener(
    new RadioGroup.OnCheckedChangeListener() {
        @Override
        public void onCheckedChanged(RadioGroup group,int checkedId) {
            RadioButton radioButton = (RadioButton)group.findViewById(checkedId);
            type_id = group.indexOfChild(radioButton);
        }
    });
groupByRadioGroup.check(R.id.studentRadioButton);
registerButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if(
            emailEditText.getText().toString().length()>0 &&
            phoneEditText.getText().toString().length()>0 &&
            nameEditText.getText().toString().length()>0 &&
            passwordEditText.getText().toString().length()>0
        )
    {

```

```

userModel = new user(
    emailEditText.getText().toString(),
    nameEditText.getText().toString(),
    phoneEditText.getText().toString(),
    passwordEditText.getText().toString(),
    type_id
);
PostData(userModel);
}
else {
    Toast.makeText(getApplicationContext(),"Enter Details
Properly",Toast.LENGTH_LONG).show();
}
});

}

private void PostData(user userModel) {
    // Write a message to the database
    FirebaseDatabase database = FirebaseDatabase.getInstance();
    DatabaseReference myRef = database.getReference();
    myRef.child("users").child(userModel.getEmail().split("@")[0]).setValue(userModel);
    Intent i = new Intent(registrationActivity.this,loginActivity.class);
    startActivity(i);
    finish();
}
}

```

MODEL-USER.JAVA

```
package com.example.newproject_testing.models;

public class user {
    private String email;
    private String name;
    private String phone;
    private String password;
    private Integer id;
    public user(String email, String name, String phone, String password, Integer id) {
        this.email = email;
        this.name = name;
        this.phone = phone;
        this.password = password;
        this.id = id;
    }

    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getPhone() {
        return phone;
    }
}
```

```
public void setPhone(String phone) {  
    this.phone = phone;  
}  
public String getPassword() {  
    return password;  
}  
public void setPassword(String password) {  
    this.password = password;  
}  
public Integer getId() {  
    return id;  
}  
public void setId(Integer id) {  
    this.id = id;  
}  
}
```

MYSHARED PREFERENCE.JAVA

```
package com.example.newproject_testing;

import android.content.Context;
import android.content.SharedPreferences;

public class MySharedPreference {
    public static String pref_name = "Mypref";
    public static String is_login = "is_login";
    public static String type_id = "type_id";
    public static String email = "email";
    SharedPreferences preferences;
    SharedPreferences.Editor editor;
    Context context;
    MySharedPreference(Context context) {
        this.context = context;
        preferences = context.getSharedPreferences(pref_name,Context.MODE_PRIVATE);
        editor = preferences.edit();
    }
    public boolean isIs_login() {
        return preferences.getBoolean(is_login,false);
    }
    public void setIs_login(boolean login) {
        editor.putBoolean(is_login,login);
        editor.apply();
        editor.commit();
    }
    public String getEmail() {
        return preferences.getString(email,"");
    }
    public void setEmail(String em) {
        editor.putString(email,em);
    }
}
```

```
editor.apply();
editor.commit();
}

public int getType_id() {
    return preferences.getInt(type_id,0);
}

public void setType_id(int id) {
    editor.putInt(type_id,id);
    editor.apply();
    editor.commit();
}

public void Logout(){
    setIs_login(false);
    editor.clear();
}

}
```

ACTIVITY_COURSE.XML

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
    tools:context=".CourseActivity">

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <androidx.appcompat.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_marginBottom="10dp"
            android:layout_width="match_parent"
            android:layout_height="wrap_content">
        </androidx.appcompat.widget.Toolbar>

        <ImageView
            android:layout_width="match_parent"
            android:layout_height="295dp"
            android:layout_margin="16dp"
            android:scaleType="centerCrop"
            android:src="@drawable/cover_page" />
    
```

```
<TextView
    android:id="@+id/registerTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="EDUCATION MANAGEMENT SYSTEM"
    android:textColor="@color/colorPrimary"
    android:textSize="20dp"
    android:layout_gravity="center"
    android:layout_marginBottom="10dp"/>

<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/recyclerView"
    tools:listitem="@layout/item_course"
    app:layoutManager="androidx.recyclerview.widget.GridLayoutManager"
    app:spanCount="3"
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>

</LinearLayout>

<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:layout_width="wrap_content"
    android:layout_margin="15dp"
    android:id="@+id/float_bt"
    android:src="@drawable/baseline_add_24"
    android:layout_gravity="bottom|end"
    android:layout_height="wrap_content">
</com.google.android.material.floatingactionbutton.FloatingActionButton>

</FrameLayout>
```

ITEM_COURSE.XML

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/card"
    android:layout_margin="5dp"
    app:cardCornerRadius="5dp">

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="12dp"
        android:layout_gravity="center"
        android:orientation="vertical">

        <ImageView
            android:id="@+id/icon"
            android:layout_width="80dp"
            android:layout_height="80dp"
            android:layout_marginBottom="16dp"
            android:scaleType="centerCrop"
            android:layout_gravity="center"
            android:src="@drawable/course" />

        <TextView
            android:id="@+id/courseCodeTextView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="24sp"
```

```
        android:scaleType="centerCrop"
        android:layout_gravity="center"
        android:text="Course - 1"
        android:textStyle="bold"/>

<TextView
    android:id="@+id/courseTitleTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="22sp"
    android:scaleType="centerCrop"
    android:layout_gravity="center"
    android:text="Course Description"/>

</LinearLayout>
</androidx.cardview.widget.CardView>
```

COURSEACTIVITY.JAVA

```
package com.example.newproject_testing;

import android.annotation.SuppressLint;
import android.app.Dialog;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.ImageView;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
import androidx.recyclerview.widget.GridLayoutManager;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import com.example.newproject_testing.models.model_Course;
import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.HashSet;
import java.util.List;
import java.util.Set;
```

```

public class CourseActivity extends AppCompatActivity {

    private RecyclerView recyclerView;
    private CourseAdapter adapter;
    private ArrayList<model_Course> courses;
    private Toolbar toolbar;
    MySharedPreference preference;
    FloatingActionButton floatBt;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_course);
        toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        preference = new MySharedPreference(this);
        courses = new ArrayList<>();
        FirebaseDatabase database = FirebaseDatabase.getInstance();
        DatabaseReference myRef = database.getReference("Course");
        recyclerView = findViewById(R.id.recyclerView);
        myRef.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                courses.clear();
                courses = new ArrayList<>();
                List<HashMap<String, Object>> hashes = (List<HashMap<String, Object>>) dataSnapshot.getValue();
                assert hashes != null;
                for(int i=0;i<hashes.size();i++) {
                    courses.add(new model_Course(hashes.get(i).get("course_id").toString(),
                        hashes.get(i).get("course_name").toString(),hashes.get(i).get("course_icon").toString()));
                }
                recyclerView.setLayoutManager(new GridLayoutManager(CourseActivity.this,2));
                adapter = new CourseAdapter(CourseActivity.this,courses);
            }
        });
    }
}

```

```

recyclerView.setAdapter(adapter);
}

@Override
public void onCancelled(@NonNull DatabaseError error) {
    Log.w("Testing", "Failed to read value.", error.toException());
}
});

}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    super.onCreateOptionsMenu(menu);
    MenuInflater inflater = new MenuInflater(this);
    inflater.inflate(R.menu.op_menu,menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {
    super.onOptionsItemSelected(item);
    if(item.getItemId() == R.id.logout){
        preference.Logout();
        Intent i = new Intent(this,logoActivity.class);
        i.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(i);
    }
    if(item.getItemId() == R.id.profile){
        Intent i = new Intent(this,ProfileActivity.class);
        startActivity(i);
    }
    if (item.getItemId() == R.id.about) {
        showCustomDialog();
        return true;
    }
    return true;
}

```

```

}

private void showCustomDialog() {
    Dialog dialog = new Dialog(this);
    dialog.setContentView(R.layout.activity_about);
    ImageView imageView = dialog.findViewById(R.id.dialog_image);
    TextView titleTextView = dialog.findViewById(R.id.dialog_title);
    TextView descriptionTextView = dialog.findViewById(R.id.dialog_description);
    imageView.setImageResource(R.drawable.sss);
    titleTextView.setText("SS ज्ञान Sync HUB");
    descriptionTextView.setText(
        "SS GYAN SYNC HUB is a cutting-edge Education Management System (EMS)  

        revolutionizing learning experiences.\n\n"
        +"It empowers educators with innovative tools for seamless course management and  

        instructional delivery.\n\n"
        +"For students, it offers a personalized learning environment with access to diverse  

        educational resources and real-time communication.\n\n"
        +"Harnessing data and analytics, it drives continuous improvement and fosters  

        collaboration, transforming education for the digital age."
    );
    dialog.show();
}
}

```

COURSEADAPTER.JAVA

```
package com.example.newproject_testing;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.cardview.widget.CardView;
import androidx.recyclerview.widget.RecyclerView;
import com.bumptech.glide.Glide;
import com.example.newproject_testing.models.model_Course;
import java.util.List;

public class CourseAdapter extends RecyclerView.Adapter<CourseAdapter.ViewHolder> {
    private List<model_Course> courses;
    private Context context;
    public CourseAdapter(Context context, List<model_Course> courses) {
        this.context = context;
        this.courses = courses;
    }
    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(context).inflate(R.layout.item_course, parent, false);
        return new ViewHolder(view);
    }
    @Override
```

```

public void onBindViewHolder(@NonNull ViewHolder holder, int position) {
    model_Course course = courses.get(position);
    holder.courseCodeTextView.setText(course.getCourseCode());
    holder.courseTitleTextView.setText(course.getCourseTitle());
    holder.cardView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent i = new Intent(context,SubjectActivity.class);
            i.putExtra("course_id",course.getCourseCode());
            i.putExtra("course_icon",course.getCourseIcon());
            context.startActivity(i);
        }
    });
}

Glide
.with(context)
.load(course.getCourseIcon())
.centerCrop()
.into(holder.icon);
}

@Override
public int getItemCount() {return courses.size();}

public static class ViewHolder extends RecyclerView.ViewHolder {
    TextView courseCodeTextView;
    TextView courseTitleTextView;
    CardView cardView;
    ImageView icon;

    public ViewHolder(@NonNull View itemView) {
        super(itemView);
        courseCodeTextView = itemView.findViewById(R.id.courseCodeTextView);
        courseTitleTextView = itemView.findViewById(R.id.courseTitleTextView);
        icon = itemView.findViewById(R.id.icon);
        cardView = itemView.findViewById(R.id.card);
    }
}

```

MODEL-COURSE.JAVA

```
package com.example.newproject_testing.models;

public class model_Course {
    private String courseCode;
    private String courseTitle;
    private String course_icon;
    public model_Course(String courseCode, String courseTitle, String course_icon) {
        this.courseCode = courseCode;
        this.courseTitle = courseTitle;
        this.course_icon = course_icon;
    }

    public String getCourseCode() {
        return courseCode;
    }
    public void setCourseCode(String courseCode) {
        this.courseCode = courseCode;
    }
    public String getCourseTitle() {
        return courseTitle;
    }
    public void setCourseTitle(String courseTitle) {
        this.courseTitle = courseTitle;
    }
    public String getCourse_icon() {
        return course_icon;
    }
    public void setCourse_icon(String course_icon) {
        this.course_icon = course_icon;
    }
}
```

ACTIVITY_ABOUT.XML

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="16dp">

    <ImageView
        android:id="@+id/dialog_image"
        android:layout_width="match_parent"
        android:layout_height="328dp"
        android:scaleType="centerCrop"
        android:layout_gravity="center"
        android:src="@drawable/sss" />

    <TextView
        android:id="@+id/dialog_title"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="18sp"
        android:textStyle="bold"
        android:paddingTop="16dp"
        android:textAlignment="center" />

    <TextView
        android:id="@+id/dialog_description"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="8dp"
        android:justificationMode="inter_word" />
</LinearLayout>
```

ACTIVITY_PROFILE.XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <EditText
        android:id="@+id/editTextEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Email" />

    <EditText
        android:id="@+id/editTextName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Name" />

    <EditText
        android:id="@+id/editTextPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Password" />

    <EditText
        android:id="@+id/editTextPhone"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Phone" />

</LinearLayout>
```

PROFILEACTIVITY.JAVA

```
package com.example.newproject_testing;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.widget.EditText;
import android.widget.Toast;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebaseio.database.DataSnapshot;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
import java.util.HashMap;

public class ProfileActivity extends AppCompatActivity {

    MySharedPreference preference;
    EditText editTextEmail, editTextName, editTextPassword, editTextPhone;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_profile);
        preference = new MySharedPreference(this);
        editTextEmail = findViewById(R.id.editTextEmail);
        editTextName = findViewById(R.id.editTextName);
        editTextPassword = findViewById(R.id.editTextPassword);
        editTextPhone = findViewById(R.id.editTextPhone);
        FirebaseDatabase database = FirebaseDatabase.getInstance();
        DatabaseReference myRef = database.getReference("users");
```

```
myRef.child(preference.getEmail().split("@")[0]).get().addOnCompleteListener(new  
OnCompleteListener<DataSnapshot>() {  
    @Override  
    public void onComplete(@NonNull Task<DataSnapshot> task) {  
        if (!task.isSuccessful()) {  
            Log.e("firebaseDATA", "Error getting data", task.getException());  
        }  
        else {  
            Log.e("firebaseDATA", String.valueOf(task.getResult().getValue()));  
            HashMap<String , Object> map = (HashMap<String, Object>)  
task.getResult().getValue();  
            String password = map.get("password").toString();  
            String email = map.get("email").toString();  
            String name = map.get("name").toString();  
            String phone = map.get("phone").toString();  
            editTextEmail.setText(email);  
            editTextName.setText(name);  
            editTextPassword.setText(password);  
            editTextPhone.setText(phone);  
        }  
    }  
});  
}  
}
```

ACTIVITY SUBJECT.XML

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    tools:context=".SubjectActivity">

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TextView
            android:id="@+id/registerTextView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="COURSES"
            android:textColor="@color/colorPrimary"
            android:textSize="20dp"
            android:layout_gravity="center"
            android:layout_marginBottom="10dp"/>

        <ImageView
            android:id="@+id/img"
            android:layout_width="match_parent"
            android:layout_height="295dp"
            android:layout_margin="16dp"
            android:scaleType="centerCrop"
            android:src="@drawable/course_material" />
    
```

```
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/recyclerView"
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>
</LinearLayout>

<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:layout_width="wrap_content"
    android:layout_margin="15dp"
    android:id="@+id/float_bt"
    android:src="@drawable/baseline_notifications_24"
    android:layout_gravity="bottom|end"
    android:layout_height="wrap_content">
</com.google.android.material.floatingactionbutton.FloatingActionButton>

</FrameLayout>
```

ITEM SUBJECT.XML

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView app:contentPadding="16dp"
    android:layout_margin="12dp"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="16dp"
    android:id="@+id/card"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <TextView
        android:id="@+id/subjectTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="18sp"
        android:text="Topics"/>

    <ImageView
        android:id="@+id/icon"
        android:layout_width="80dp"
        android:layout_height="80dp"
        android:layout_marginBottom="16dp"
        android:scaleType="centerCrop"
        android:layout_gravity="center"
        android:src="@drawable/course" />

</androidx.cardview.widget.CardView>
```

SUBJECTACTIVITY.JAVA

```
package com.example.newproject_testing;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.ImageView;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import com.bumptech.glide.Glide;
import com.example.newproject_testing.models.model_Subject;
import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.HashSet;
import java.util.List;
import java.util.Set;

public class SubjectActivity extends AppCompatActivity {

    private RecyclerView recyclerView;
    private SubjectAdapter adapter;
    private List<model_Subject> subjects;
    private String course_id, course_icon;
    FloatingActionButton floatingActionButton;
```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_subject);
    ImageView imageView = findViewById(R.id.img);
    floatingActionButton = findViewById(R.id.float_bt);
    course_id = getIntent().getStringExtra("course_id");
    course_icon = getIntent().getStringExtra("course_icon");
    Glide
        .with(this)
        .load(course_icon)
        .centerCrop()
        .into(imageView);
    floatingActionButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent i = new Intent(SubjectActivity.this,NotificationActivity.class);
            startActivity(i);
        }
    });
    if(course_id!=null) {
        subjects = new ArrayList<>();
        FirebaseDatabase database = FirebaseDatabase.getInstance();
        DatabaseReference myRef = database.getReference("Subject");
        recyclerView = findViewById(R.id.recyclerView);
        myRef.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                subjects.clear();
                subjects = new ArrayList<>();
                List<HashMap<String, Object>> hashes = (List<HashMap<String, Object>>)
                dataSnapshot.getValue();
                Log.e("TestingFirebase", dataSnapshot.getValue().toString());
            }
        });
    }
}

```

```
assert hashes != null;

for(int i=0;i<hashes.size();i++) {
    if(course_id.equals(hashes.get(i).get("course_id").toString())) {
        subjects.add(new model_Subject(
            hashes.get(i).get("subject_id").toString(),
            hashes.get(i).get("subject_name").toString(),
            hashes.get(i).get("course_id").toString(),
            hashes.get(i).get("subject_icon").toString()
        ));
    }
}

recyclerView.setLayoutManager(new LinearLayoutManager(SubjectActivity.this));
adapter = new SubjectAdapter(SubjectActivity.this,subjects);
recyclerView.setAdapter(adapter);
}

@Override

public void onCancelled(@NonNull DatabaseError error) {
    Log.e("TestingFirebase", "Failed to read value.", error.toException());
}

});
```

SUBJECTADAPTER.JAVA

```
package com.example.newproject_testing;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.cardview.widget.CardView;
import androidx.recyclerview.widget.RecyclerView;
import com.bumptech.glide.Glide;
import com.example.newproject_testing.models.model_Subject;
import java.util.List;

public class SubjectAdapter extends RecyclerView.Adapter<SubjectAdapter.ViewHolder> {

    private List<model_Subject> subjects;
    private Context context;
    public SubjectAdapter(Context context, List<model_Subject> subjects) {
        this.context = context;
        this.subjects = subjects;
    }
    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.item_subject, parent,
false);
        return new ViewHolder(view);
    }
    @Override
```

```

public void onBindViewHolder(@NonNull ViewHolder holder, int position) {
    model_Subject subject = subjects.get(position);
    holder.subjectTitleTextView.setText(subject.getSubjectName());
    holder.cardView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent i = new Intent(context, ActionsActivity.class);
            i.putExtra("subject_id", subject.getSubjectId());
            i.putExtra("course_icon", subject.getSubjectIcon());
            context.startActivity(i);
        }
    });
}

Glide
    .with(context)
    .load(subject.getSubjectIcon())
    .centerCrop()
    .into(holder.icon);
}

@Override
public int getItemCount() {
    return subjects.size();
}

public static class ViewHolder extends RecyclerView.ViewHolder {
    ImageView icon;
    TextView subjectCodeTextView;
    TextView subjectTitleTextView;
    CardView cardView;
    public ViewHolder(@NonNull View itemView) {
        super(itemView);
        subjectTitleTextView = itemView.findViewById(R.id.subjectTextView);
        icon = itemView.findViewById(R.id.icon);
        cardView = itemView.findViewById(R.id.card);
    }
}

```

MODEL-SUBJECT.XML

```
package com.example.newproject_testing.models;

public class model_Subject {
    private String subjectId;
    private String subjectName;
    private String courseId;
    private String subject_icon;
    public model_Subject(String subjectId, String subjectName, String courseId, String
subject_icon) {
        this.subjectId = subjectId;
        this.subjectName = subjectName;
        this.courseId = courseId;
        this.subject_icon = subject_icon;
    }

    public String getSubjectId() {
        return subjectId;
    }
    public void setSubjectId(String subjectId) {
        this.subjectId = subjectId;
    }
    public String getSubjectName() {
        return subjectName;
    }
    public void setSubjectName(String subjectName) {
        this.subjectName = subjectName;
    }
    public String getCourseId() {
        return courseId;
    }
}
```

```
public void setCourseId(String courseId) {  
    this.courseId = courseId;  
}  
public String getSubject_icon() {  
    return subject_icon;  
}  
public void setSubject_icon(String subject_icon) {  
    this.subject_icon = subject_icon;  
}  
}
```

ACTIVITY_ACTION.XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    tools:context=".ActionsActivity">

    <TextView
        android:id="@+id/registerTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="COURSE MATERIALS"
        android:textColor="@color/colorPrimary"
        android:textSize="20dp"
        android:layout_gravity="center"
        android:layout_marginBottom="10dp"/>

    <ImageView
        android:id="@+id/img"
        android:layout_width="match_parent"
        android:layout_height="314dp"
        android:layout_marginBottom="16dp"
        android:scaleType="centerCrop"
        android:src="@drawable/course_material" />
```

```
<androidx.recyclerview.widget.RecyclerView  
    android:id="@+id/recyclerView"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"/>  
  
</LinearLayout>
```

ITEM_ACTION.XML

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView app:contentPadding="16dp"
    android:layout_margin="12dp"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="16dp"
    android:id="@+id/card"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <TextView
        android:id="@+id/actionTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="18sp"
        android:text="Topics"/>

</androidx.cardview.widget.CardView>
```

ACTIONACTIVITY.JAVA

```
package com.example.newproject_testing;

import android.os.Bundle;
import android.util.Log;
import android.widget.ImageView;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import com.bumptech.glide.Glide;
import com.example.newproject_testing.models.model_Action;
import com.example.newproject_testing.models.model_Subject;
import com.google.firebaseio.database.DataSnapshot;
import com.google.firebaseio.database.DatabaseError;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
import com.google.firebaseio.database.ValueEventListener;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

public class ActionsActivity extends AppCompatActivity {
    private RecyclerView recyclerView;
    private ActionAdapter adapter;
    private List<model_Action> actions;
    private String subject_id, course_icon;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_actions);
        ImageView imageView = findViewById(R.id.img);
```

```
course_icon = getIntent().getStringExtra("course_icon");
Glide
    .with(this)
    .load(course_icon)
    .centerCrop()
    .into(imageView);

recyclerView = findViewById(R.id.recyclerView);
subject_id = getIntent().getStringExtra("subject_id");
Log.e("TestingFirebase",subject_id);
actions = new ArrayList<>();
actions.add(new model_Action("Learn PDFs",1));
actions.add(new model_Action("Video Tutorials",2));
actions.add(new model_Action("Quiz",3));
adapter = new ActionAdapter(ActionsActivity.this,actions,subject_id);
recyclerView.setLayoutManager(new
LinearLayoutManager(this,LinearLayoutManager.VERTICAL,false));
recyclerView.setAdapter(adapter);
}
}
```

ACTIONADAPTER.JAVA

```
package com.example.newproject_testing;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.cardview.widget.CardView;
import androidx.recyclerview.widget.RecyclerView;
import com.example.newproject_testing.models.model_Action;
import java.util.List;

public class ActionAdapter extends RecyclerView.Adapter<ActionAdapter.ViewHolder> {

    private List<model_Action> actions;
    private Context context;
    private String subject_id;

    public ActionAdapter(Context context, List<model_Action> actions, String subject_id) {
        this.actions = actions;
        this.context=context;
        this.subject_id = subject_id;
    }

    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.item_action, parent, false);
        return new ViewHolder(view);
    }

    @Override
```

```

public void onBindViewHolder(@NonNull ViewHolder holder, int position) {
    model_Action action = actions.get(position);
    holder.actionTextView.setText(action.getActionName());
    holder.cardView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if(action.getActionId() == 1) {
                Intent i = new Intent(context, PDFActivity.class);
                i.putExtra("subject_id", subject_id);
                context.startActivity(i);
            }
            if(action.getActionId() == 2) {
                Intent i = new Intent(context, VideoActivity.class);
                i.putExtra("subject_id", subject_id);
                context.startActivity(i);
            }
            if(action.getActionId() == 3) {
                Intent i = new Intent(context, QuizActivity.class);
                i.putExtra("subject_id", subject_id);
                context.startActivity(i);
            }
        }
    });
}
@Override
public int getItemCount() {
    return actions.size();
}
public static class ViewHolder extends RecyclerView.ViewHolder {
    TextView actionTextView;
    CardView cardView;
    public ViewHolder(@NonNull View itemView) {
        super(itemView);
    }
}

```

```
    actionTextView = itemView.findViewById(R.id.actionTextView);
    cardView = itemView.findViewById(R.id.card);
}
}
}
```

MODEL-ACTION.JAVA

```
package com.example.newproject_testing.models;

public class model_Action {
    private String actionPerformed;
    private int actionId;
    public model_Action(String actionPerformed, Integer actionId) {
        this.actionPerformed = actionPerformed;
        this.actionId = actionId;
    }

    public int getActionId() {
        return actionId;
    }
    public void setActionId(int actionId) {
        this.actionId = actionId;
    }
    public String actionPerformed() {
        return actionPerformed;
    }
    public void actionPerformed(String actionPerformed) {
        this.actionPerformed = actionPerformed;
    }
}
```

ACTIVITY_PDF.XML

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    tools:context=".PDFActivity">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:id="@+id/addPDFButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentEnd="true"
        android:layout_margin="16dp"
        android:src="@drawable/baseline_add_24" />

</RelativeLayout>
```

ITEM_PDF.XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="16dp">

    <TextView
        android:id="@+id/pdfNameTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="18sp"
        android:text="PDFs"/>

</LinearLayout>
```

ACTIVTIY_PDF_LAYOUT.XML

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".CourseActivity">

    <androidx.recyclerview.widget.RecyclerView
        tools:listitem="@layout/item_pdf"
        android:id="@+id/recyclerView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:layout_width="wrap_content"
        android:layout_margin="15dp"
        android:src="@drawable/baseline_add_24"
        android:layout_gravity="bottom|end"
        android:layout_height="wrap_content">
    </com.google.android.material.floatingactionbutton.FloatingActionButton>

</FrameLayout>
```

ACTIVITY_PDF_UPDATE.XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp" >

    <EditText
        android:id="@+id/editText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Text"/>

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Click Me"/>

</LinearLayout>
```

PDFACTIVITY.JAVA

```
package com.example.newproject_testing;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.webkit.WebSettings;
import android.webkit.WebView;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import com.example.newproject_testing.models.pdf;
import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

public class PDFActivity extends AppCompatActivity {
    private RecyclerView recyclerView;
    private PDFAdapter adapter;
    private List<pdf> pdfs;
    private List<pdf> Allpdfs;
    String subject_id;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_pdf);
recyclerView = findViewById(R.id.recycler_view);
recyclerView.setLayoutManager(new LinearLayoutManager(this));
subject_id = getIntent().getStringExtra("subject_id");
pdfs = new ArrayList<>();
Allpdfs = new ArrayList<>();
FirebaseDatabase database = FirebaseDatabase.getInstance();
DatabaseReference myRef = database.getReference("Data_PDF");
myRef.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        pdfs.clear();
        Allpdfs.clear();
        pdfs = new ArrayList<>();
        Allpdfs = new ArrayList<>();
        List<HashMap<String, Object>> hashes = (List<HashMap<String, Object>>)
        dataSnapshot.getValue();
        assert hashes != null;
        for(int i=0;i<hashes.size();i++) {
            Allpdfs.add(new
                pdf(Integer.parseInt(hashes.get(i).get("subject_id").toString()),hashes.get(i).get("pdfs_link").toString()));
            if(subject_id.equals(hashes.get(i).get("subject_id").toString()) &&
            !hashes.get(i).get("pdfs_link").toString().equals("NA")) {
                pdfs.add(new
                    pdf(Integer.parseInt(hashes.get(i).get("subject_id").toString()),hashes.get(i).get("pdfs_link").toString()));
            }
        }
        adapter = new PDFAdapter(PDFActivity.this,pdfs,Allpdfs);
        recyclerView.setAdapter(adapter);
        Log.e("POSITIONADAPTER",Allpdfs.size()+"");
    }
}

```

```

    }

    @Override
    public void onCancelled(@NonNull DatabaseError error) {
        Log.w("Testing", "Failed to read value.", error.toException());
    }
});

pdfs.add(new pdf(1,"PDF 1"));
pdfs.add(new pdf(1,"PDF 2"));
pdfs.add(new pdf(1,"PDF 3"));

FloatingActionButton addPDFButton = findViewById(R.id.addPDFButton);
addPDFButton.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {
        Intent in = new Intent(PDFActivity.this,PDFUpdate.class);
        in.putExtra("subject_id",Integer.parseInt(subject_id));
        in.putExtra("pos",Allpdfs.size());
        in.putExtra("link","");
        startActivity(in);
    }
});
}

private class Builder {

}

private class ViewerConfig {

}
}

```

PDFADAPTER.JAVA

```
package com.example.newproject_testing;

import android.content.Context;
import android.content.Intent;
import android.net.Uri;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;
import com.example.newproject_testing.models.pdf;
import com.pdftron.pdf.config.ViewerConfig;
import com.pdftron.pdf.controls.DocumentActivity;
import java.util.List;

public class PDFAdapter extends RecyclerView.Adapter<PDFAdapter.ViewHolder> {
    private List<pdf> pdfs;
    private List<pdf> allpdfs;
    private Context context;
    private MySharedPreference preference;
    private int pos;
    public PDFAdapter(Context context, List<pdf> pdfs, List<pdf> allpdfs) {
        this.pdfs = pdfs;
        this.context = context;
        this.allpdfs= allpdfs;
        preference = new MySharedPreference(context);
    }
    @NonNull
    @Override
```

```

public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
    View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.item_pdf, parent,
false);
    return new ViewHolder(view);
}

@Override
public void onBindViewHolder(@NonNull ViewHolder holder, int position) {
    pdf pdf = pdfs.get(position);
    holder.pdfNameTextView.setText(pdf.getPdfs_link());
    holder.pdfNameTextView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if(preference.getType_id()==1){ // Set the cache location using the config to store the
cache file
                ViewerConfig.Builder builder = new ViewerConfig.Builder();
                ViewerConfig config = builder
                    .fullscreenModeEnabled(true)
                    .multiTabEnabled(true)
                    .documentEditingEnabled(true)
                    .longPressQuickMenuEnabled(true)
                    .toolbarTitle("Simple Reader")
                    .showSearchView(true)
                    .build();
                final Uri uri = Uri.parse(pdf.getPdfs_link());
                Intent intent = DocumentActivity.IntentBuilder.fromActivityClass(context,
DocumentActivity.class)
                    .withUri(uri)
                    .usingConfig(config)
                    .build();
                context.startActivity(intent);
            }
        }
    for(int i=0;i<allpdfs.size();i++){
        if(allpdfs.get(i).getPdfs_link().equals(pdf.getPdfs_link())){

```

```

        pos = i;
    }
}

if (preference.getType_id() == 0){
    Intent in = new Intent(context, PDFUpdate.class);
    in.putExtra("subject_id", pdf.getSubject_id());
    in.putExtra("pos", pos);
    in.putExtra("link", pdf.getPdfs_link());
    context.startActivity(in);
}
}

});

}

@Override
public int getItemCount() {
    return pdfs.size();
}

public static class ViewHolder extends RecyclerView.ViewHolder {
    TextView pdfNameTextView;

    public ViewHolder(@NonNull View itemView) {
        super(itemView);
        pdfNameTextView = itemView.findViewById(R.id.pdfNameTextView);
    }
}
}

```

PDFUPDATE.JAVA

```
package com.example.newproject_testing;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import com.example.newproject_testing.models.pdf;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;

public class PDFUpdate extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_pdf_update);
        EditText editText = findViewById(R.id.editText);
        Button button = findViewById(R.id.button);
        Integer subject_id = getIntent().getIntExtra("subject_id",0);
        Integer pos = getIntent().getIntExtra("pos",0);
        String link = getIntent().getStringExtra("link");
        editText.setText(link);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                FirebaseDatabase database = FirebaseDatabase.getInstance();
                DatabaseReference myRef = database.getReference();
                pdf pdfmodel = new pdf(subject_id,editText.getText().toString());
                myRef.child("Data_PDF").child(String.valueOf(pos)).setValue(pdfmodel);
            }
        });
    }
}
```

MODEL-PDF.JAVA

```
package com.example.newproject_testing.models;

public class pdf {
    private Integer subject_id;
    private String pdfs_link;
    private final static long serialVersionUID = 7122027055016191247L;
    public pdf(Integer subject_id, String pdfs_link) {
        this.subject_id = subject_id;
        this.pdfs_link = pdfs_link;
    }

    public Integer getSubject_id() {
        return subject_id;
    }
    public void setSubject_id(Integer subject_id) {
        this.subject_id = subject_id;
    }
    public String getPdfs_link() {
        return pdfs_link;
    }
    public void setPdfs_link(String pdfs_link) {
        this.pdfs_link = pdfs_link;
    }
}
```

ACTIVITY_VIDEO.XML

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".VideoActivity">
    <WebView
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    </WebView>

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerView"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    </androidx.recyclerview.widget.RecyclerView>
    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:id="@+id/addPDFButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom|end"
        android:layout_margin="16dp"
        android:src="@drawable/baseline_add_24" />
</FrameLayout>
```

ITEM_VIDEO.XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="16dp">

    <TextView
        android:id="@+id/videoNameTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="18sp"
        android:text="VIDEO"/>

</LinearLayout>
```

ACTIVITY_VIDEO_LAYOUT.XML

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".CourseActivity">

    <androidx.recyclerview.widget.RecyclerView
        tools:listitem="@layout/item_video"
        android:id="@+id/recyclerView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:layout_width="wrap_content"
        android:layout_margin="15dp"
        android:src="@drawable/baseline_add_24"
        android:layout_gravity="bottom|end"
        android:layout_height="wrap_content">
    </com.google.android.material.floatingactionbutton.FloatingActionButton>

</FrameLayout>
```

ACTIVITY_VIDEO_UPDATE.XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp" >

    <EditText
        android:id="@+id/editText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Text"/>

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Click Me"/>

</LinearLayout>
```

VIDEOACTIVITY.JAVA

```
package com.example.newproject_testing;

import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import com.example.newproject_testing.models.model_Course;
import com.example.newproject_testing.models.pdf;
import com.example.newproject_testing.models.video;
import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

public class VideoActivity extends AppCompatActivity {
    private List<video> videos;
    private List<video> Allvideos;
    String subject_id;
```

```

WebView wv;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_video);
    subject_id = getIntent().getStringExtra("subject_id");
    RecyclerView recyclerView = findViewById(R.id.recyclerView);
    recyclerView.setLayoutManager(new LinearLayoutManager(this));
    videos = new ArrayList<>();
    FirebaseDatabase database = FirebaseDatabase.getInstance();
    DatabaseReference myRef = database.getReference("Data_VIDEO");
    myRef.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            videos.clear();
            videos = new ArrayList<>();
            Allvideos = new ArrayList<>();
            List<HashMap<String, Object>> hashes = (List<HashMap<String, Object>>)
            dataSnapshot.getValue();
            assert hashes != null;
            for(int i=0;i<hashes.size();i++) {
                if(subject_id.equals(hashes.get(i).get("subject_id").toString()) &&
                !hashes.get(i).get("videos_link").toString().equals("NA")) {
                    videos.add(new
                    video(hashes.get(i).get("subject_id").toString(),hashes.get(i).get("videos_link").toString()));
                }
                Allvideos.add(new
                video(hashes.get(i).get("subject_id").toString(),hashes.get(i).get("videos_link").toString()));
            }
            VideoAdapter adapter = new VideoAdapter(VideoActivity.this,videos,Allvideos);
            recyclerView.setAdapter(adapter);
            Log.e("TestingFirebase", dataSnapshot.getValue().toString());
        }
    });
}

```

```
@Override
public void onCancelled(@NonNull DatabaseError error) {
    Log.w("Testing", "Failed to read value.", error.toException());
}

FloatingActionButton addPDFButton = findViewById(R.id.addPDFButton);
addPDFButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent in = new Intent(VideoActivity.this, VideoUpdate.class);
        in.putExtra("subject_id", subject_id);
        in.putExtra("pos", Allvideos.size());
        in.putExtra("link", "");
        startActivity(in);
    }
});

}
}
```

VIDEOADAPTER.JAVA

```
package com.example.newproject_testing;

import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;
import com.example.newproject_testing.models.video;
import java.util.List;

public class VideoAdapter extends RecyclerView.Adapter<VideoAdapter.ViewHolder> {
    private List<video> videos;
    private List<video> allvideos;
    private Context context;
    private Context mActivityContext;
    private MySharedPreference preference;
    private int pos;
    public VideoAdapter(Context context, List<video> videos, List<video> allvideos) {
        this.videos = videos;
        this.allvideos = allvideos;
        this.context = context;
        mActivityContext = context;
        preference = new MySharedPreference(context);
    }
    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.item_video, parent, false);
        return new ViewHolder(view);
    }
    @Override
    public void onBindViewHolder(ViewHolder holder, int position) {
        holder.bind(videos.get(position));
    }
    @Override
    public int getItemCount() {
        return videos.size();
    }
    class ViewHolder extends RecyclerView.ViewHolder {
        private TextView title;
        private TextView desc;
        private TextView duration;
        private ImageView thumbnail;
        public ViewHolder(View itemView) {
            super(itemView);
            title = itemView.findViewById(R.id.title);
            desc = itemView.findViewById(R.id.desc);
            duration = itemView.findViewById(R.id.duration);
            thumbnail = itemView.findViewById(R.id.thumbnail);
        }
        void bind(video video) {
            title.setText(video.getTitle());
            desc.setText(video.getDesc());
            duration.setText(video.getDuration());
            Glide.with(itemView.getContext()).load(video.getThumbnail()).into(thumbnail);
        }
    }
}
```

```

@Override
public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
    View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.item_video, parent,
false);
    return new ViewHolder(view);
}

@Override
public void onBindViewHolder(@NonNull ViewHolder holder, int position) {
    video video = videos.get(position);
    holder.videoNameTextView.setText(video.getVideos_link());
    holder.videoNameTextView.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View view) {
            if (preference.getType_id()==1) {
                openWebView(video.getVideos_link());
            }
            for(int i=0;i<allvideos.size();i++){
                if(allvideos.get(i).getVideos_link().equals(video.getVideos_link())){
                    pos = i;
                }
            }
            if (preference.getType_id()==0){
                Intent in = new Intent(context,VideoUpdate.class);
                in.putExtra("subject_id",video.getSubject_id());
                in.putExtra("pos",pos);
                in.putExtra("link",video.getVideos_link());
                context.startActivity(in);
            }
        }
    });
}

@Override
public int getItemCount() {

```

```
        return videos.size();
    }

    private void openWebView(String link) {
        WebView theWebPage = new WebView(mActivityContext);
        theWebPage.getSettings().setJavaScriptEnabled(true);
        theWebPage.getSettings().setPluginState(WebSettings.PluginState.ON);
        if (mActivityContext instanceof Activity) {
            ((Activity) mActivityContext).setContentView(theWebPage);
            ((Activity) mActivityContext).finish();
        }
        theWebPage.loadUrl(link);
    }

    public static class ViewHolder extends RecyclerView.ViewHolder {
        TextView videoNameTextView;

        public ViewHolder(@NonNull View itemView) {
            super(itemView);
            videoNameTextView = itemView.findViewById(R.id.videoNameTextView);
        }
    }
}
```

VIDEOUPDATE.JAVA

```
package com.example.newproject_testing;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import androidx.appcompat.app.AppCompatActivity;
import com.example.newproject_testing.models.pdf;
import com.example.newproject_testing.models.video;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;

public class VideoUpdate extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_video_update);
        EditText editText = findViewById(R.id.editText);
        Button button = findViewById(R.id.button);
        String subject_id = getIntent().getStringExtra("subject_id");
        Integer pos = getIntent().getIntExtra("pos",0);
        String link = getIntent().getStringExtra("link");
        editText.setText(link);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                FirebaseDatabase database = FirebaseDatabase.getInstance();
                DatabaseReference myRef = database.getReference();
                video vid = new video(subject_id,editText.getText().toString());
                myRef.child("Data_VIDEO").child(String.valueOf(pos)).setValue(vid);
            } });
    }
}
```

MODEL-VIDEO.JAVA

```
package com.example.newproject_testing.models;

public class video {
    private String subject_id;
    private String videos_link;
    public video(String subject_id, String videos_link) {
        this.subject_id = subject_id;
        this.videos_link = videos_link;
    }

    public String getSubject_id() {
        return subject_id;
    }
    public void setSubject_id(String subject_id) {
        this.subject_id = subject_id;
    }
    public String getVideos_link() {
        return videos_link;
    }
    public void setVideos_link(String videos_link) {
        this.videos_link = videos_link;
    }
}
```

ACTIVTY_MCQ.XML

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    android:padding="16dp">

    <RelativeLayout
        android:visibility="gone"
        android:id="@+id/relative"
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <androidx.cardview.widget.CardView
            android:id="@+id/questionCardView"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="16dp"
            android:padding="16dp"
            android:elevation="4dp"
            android:background="@color/cardview_light_background">

            <TextView
                android:id="@+id/questionTextView"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Question"
                android:textSize="18sp"/>
        </androidx.cardview.widget.CardView>
    
```

```
<RadioGroup
    android:id="@+id/optionsLayout"
    android:layout_below="@+id/questionCardView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <RadioButton
        android:id="@+id/option1RadioButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Option 1"
        android:textSize="16sp"/>

    <RadioButton
        android:id="@+id/option2RadioButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Option 2"
        android:textSize="16sp"/>

    <RadioButton
        android:id="@+id/option3RadioButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Option 3"
        android:textSize="16sp"/>

    <RadioButton
        android:id="@+id/option4RadioButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Option 4"
        android:textSize="16sp"/>

</RadioGroup>
```

```
<Button  
    android:id="@+id/submitButton"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Submit"  
    android:layout_below="@+id/optionsLayout"  
    android:layout_centerHorizontal="true"  
    android:layout_marginTop="16dp"/>  
</RelativeLayout>  
  
<FrameLayout  
    android:id="@+id/frame"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">  
  
<androidx.recyclerview.widget.RecyclerView  
    android:id="@+id/recycler"  
    tools:listitem="@layout/item_quiz_card"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">  
</androidx.recyclerview.widget.RecyclerView>  
  
<com.google.android.material.floatingactionbutton.FloatingActionButton  
    android:layout_width="wrap_content"  
    android:layout_margin="15dp"  
    android:id="@+id/float_bt"  
    android:src="@drawable/baseline_add_24"  
    android:layout_gravity="bottom|end"  
    android:layout_height="wrap_content">  
</com.google.android.material.floatingactionbutton.FloatingActionButton>  
  
</FrameLayout>  
</RelativeLayout>
```

ITEM QUIZ.XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <EditText
        android:id="@+id/editText1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="EditText 1"/>

    <EditText
        android:id="@+id/editText2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="EditText 2"/>

    <EditText
        android:id="@+id/editText3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="EditText 3"/>

    <EditText
        android:id="@+id/editText4"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="EditText 4"/>
```

```
<EditText  
    android:id="@+id/editText5"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:hint="EditText 5"/>
```

```
<EditText  
    android:id="@+id/editText6"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:hint="EditText 6"/>
```

```
<Button  
    android:id="@+id/button"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Submit" />
```

```
</LinearLayout>
```

ITEM QUIZ CARD.XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <androidx.cardview.widget.CardView
        android:id="@+id/card"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="16dp"
        android:padding="16dp"
        android:foreground="?android:attr/selectableItemBackground">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical">

            <TextView
                android:id="@+id/textView1"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Text View 1" />

            <TextView
                android:id="@+id/textView2"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Text View 2" />
        
```

```
<TextView
    android:id="@+id/textView3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Text View 3" />

<TextView
    android:id="@+id/textView4"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Text View 4" />

<TextView
    android:id="@+id/textView5"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Text View 5" />

<TextView
    android:id="@+id/textView6"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Text View 6" />

</LinearLayout>
</androidx.cardview.widget.CardView>
</LinearLayout>
```

QUIZACTIVITY.JAVA

```
package com.example.newproject_testing;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import android.app.AlertDialog;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.Button;
import android.widget.FrameLayout;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.RelativeLayout;
import android.widget.TextView;
import com.example.newproject_testing.models.mcq;
import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
```

```
public class QuizActivity extends AppCompatActivity {  
    String subject_id;  
    private List<mcq> mcqs;  
    private List<mcq> tmcqs;  
    TextView questionTextView;  
    RadioButton  
    option1RadioButton,option2RadioButton,option3RadioButton,option4RadioButton;  
    Button submitButton;  
    int pos;  
    List<Integer> correctAns;  
    int selectedAns;  
    RadioGroup optionsLayout;  
    AlertDialog dialog;  
    AlertDialog.Builder builder;  
    FrameLayout frame;  
    RelativeLayout relative;  
    MySharedPreference preference;  
    RecyclerView recyclerView;  
    FloatingActionButton float_bt;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_mcq);  
        frame = findViewById(R.id.frame);  
        relative = findViewById(R.id.relative);  
        questionTextView = findViewById(R.id.questionTextView);  
        option1RadioButton = findViewById(R.id.option1RadioButton);  
        option2RadioButton = findViewById(R.id.option2RadioButton);  
        option3RadioButton = findViewById(R.id.option3RadioButton);  
        option4RadioButton = findViewById(R.id.option4RadioButton);  
        submitButton = findViewById(R.id.submitButton);  
        optionsLayout = findViewById(R.id.optionsLayout);  
        float_bt = findViewById(R.id.float_bt);  
    }  
}
```

```

recyclerView = findViewById(R.id.recycler);
subject_id = getIntent().getStringExtra("subject_id");
mcqs = new ArrayList<>();
correctAns = new ArrayList<>();
preference = new MySharedPreference(this);
if(preference.getType_id() == 0){
    relative.setVisibility(View.GONE);
    frame.setVisibility(View.VISIBLE);
}
else{
    frame.setVisibility(View.GONE);
    relative.setVisibility(View.VISIBLE);
}
FirebaseDatabase database = FirebaseDatabase.getInstance();
DatabaseReference myRef = database.getReference("MCQ");
myRef.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        mcqs.clear();
        mcqs = new ArrayList<>();
        tmcqs = new ArrayList<>();
        List<HashMap<String, Object>> hashes = (List<HashMap<String, Object>>)dataSnapshot.getValue();
        assert hashes != null;
        for(int i=0;i<hashes.size();i++) {
            tmcqs.add(new mcq(Integer.parseInt(hashes.get(i).get("subject_id").toString()),
                hashes.get(i).get("question").toString(),
                hashes.get(i).get("option_1").toString(),
                hashes.get(i).get("option_2").toString(),
                hashes.get(i).get("option_3").toString(),
                hashes.get(i).get("option_4").toString(),
                hashes.get(i).get("correct").toString()));
        }
    }
})

```

```

for(int i = 1;i<tmcqs.size();i++){
    if(subject_id.equals(tmcqs.get(i).getSubject_id().toString())) {
        mcqs.add(new mcq(
            tmcqs.get(i).getSubject_id(),
            tmcqs.get(i).getQuestion().toString(),
            tmcqs.get(i).getOption_1().toString(),
            tmcqs.get(i).getOption_2().toString(),
            tmcqs.get(i).getOption_3().toString(),
            tmcqs.get(i).getOption_4().toString(),
            tmcqs.get(i).getCorrect().toString()
        ));
    }
}
if(preference.getType_id()==1) {
    Log.e("TestingFirebase", mcqs.toString());
    pos = mcqs.size();
    Log.e("TESTINGMCQ",tmcqs.size()+"");
    Log.e("TESTINGMCQ",mcqs.size()+"");
    setData(pos-1);
    pos--;
}
else{
    recyclerView.setLayoutManager(new LinearLayoutManager(QuizActivity.this));
    QuizAdapter adapter = new QuizAdapter(mcqs, QuizActivity.this);
    recyclerView.setAdapter(adapter);
}
}

@Override
public void onCancelled(@NonNull DatabaseError error) {
    Log.w("Testing", "Failed to read value.", error.toException());
}
});

float_bt.setOnClickListener(new View.OnClickListener() {

```

```

@Override
public void onClick(View view) {
    Intent i = new Intent(QuizActivity.this, QuizeUpdate.class);
    i.putExtra("data", new mcq(Integer.parseInt(subject_id)));
    i.putExtra("pos", -1);
    i.putExtra("size", tmcqs.size());
    startActivity(i);
}
});

optionsLayout.setOnCheckedChangeListener(
    new RadioGroup.OnCheckedChangeListener() {
        @Override
        public void onCheckedChanged(RadioGroup group,
                                    int checkedId)
        {
            RadioButton radioButton
                = (RadioButton)group
                .findViewById(checkedId);
            int idx = group.indexOfChild(radioButton);
            selectedAns = idx+1;
        }
    });
submitButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        checkscore(pos);
        pos--;
        if(pos>=0) {
            setData(pos);
        }
        else{
            Log.e("SCORES",correctAns.size()+"");
        }
    }
});

```

```

        ScoreDialog(QuizActivity.this);
    }
}
});
}

private void ScoreDialog(Context context) {
    View view = LayoutInflater.from(context).inflate(R.layout.activity_score, null);
    TextView textViewScores = view.findViewById(R.id.textViewScores);
    textViewScores.setText(correctAns.size()+"");
    Button buttonDismiss = view.findViewById(R.id.buttonDismiss);
    buttonDismiss.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            dialog.dismiss();
        }
    });
    builder = new AlertDialog.Builder(context);
    builder.setView(view).setTitle("Exam Scores")..setCancelable(false);
    dialog = builder.create();
    dialog.show();
}

private void checkscore(int pos) {
    String[] id = mcqs.get(pos).getCorrect().split("_");
    Log.e("SCORES",mcqs.get(pos).getCorrect() + " * "+Integer.parseInt(id[1])+" * "+selectedAns);
    if(Integer.parseInt(id[1]) == selectedAns){
        correctAns.add(selectedAns);
    }
    optionsLayout.clearCheck();
}

private void setData(int pos) {
    Log.e("TESTINGMCQ",pos+" "+mcqs.get(pos).getQuestion());
    questionTextView.setText(mcqs.get(pos).getQuestion());
}

```

```
option1RadioButton.setText(mcqs.get(pos).getOption_1());
option2RadioButton.setText(mcqs.get(pos).getOption_2());
option3RadioButton.setText(mcqs.get(pos).getOption_3());
option4RadioButton.setText(mcqs.get(pos).getOption_4());

}

}
```

QUIZADAPTER.JAVA

```
package com.example.newproject_testing;

import android.content.Context;
import android.content.Intent;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.cardview.widget.CardView;
import androidx.recyclerview.widget.RecyclerView;
import com.example.newproject_testing.models.mcq;
import java.util.List;

public class QuizAdapter extends RecyclerView.Adapter<QuizAdapter.MyViewHolder> {

    List<mcq> mcqList;
    Context context;

    public QuizAdapter(List<mcq> mcqList, Context context) {
        this.mcqList = mcqList;
        this.context = context;
    }

    @NonNull
    @Override
    public MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.item_quiz_card,
parent, false);

        return new QuizAdapter.MyViewHolder(view);
    }

    @Override
```

```
public void onBindViewHolder(@NonNull MyViewHolder holder, int position) {
    holder.textView1.setText(mcqList.get(position).getQuestion());
    holder.textView2.setText(mcqList.get(position).getOption_1());
    holder.textView3.setText(mcqList.get(position).getOption_2());
    holder.textView4.setText(mcqList.get(position).getOption_3());
    holder.textView5.setText(mcqList.get(position).getOption_4());
    holder.textView6.setText(mcqList.get(position).getCorrect());
    holder.card.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent i = new Intent(context, QuizeUpdate.class);
            i.putExtra("data", mcqList.get(position));
            i.putExtra("pos", position);
            context.startActivity(i);
        }
    });
}
@Override
public int getItemCount() {
    return mcqList.size();
}
public static class MyViewHolder extends RecyclerView.ViewHolder {
    TextView textView1, textView2, textView3, textView4, textView5, textView6;
    CardView card;
    public MyViewHolder(@NonNull View itemView) {
        super(itemView);
        textView1 = itemView.findViewById(R.id.textView1);
        textView2 = itemView.findViewById(R.id.textView2);
        textView3 = itemView.findViewById(R.id.textView3);
        textView4 = itemView.findViewById(R.id.textView4);
        textView5 = itemView.findViewById(R.id.textView5);
        textView6 = itemView.findViewById(R.id.textView6);
        card = itemView.findViewById(R.id.card); } } }
```

QUIZUPDATE.JAVA

```
package com.example.newproject_testing;

import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Build;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import com.example.newproject_testing.models.mcq;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;

public class QuizeUpdate extends AppCompatActivity {
    EditText editText1, editText2, editText3, editText4, editText5, editText6;
    Button button;
    mcq data;
    int pos,size;
    @RequiresApi(api = Build.VERSION_CODES.TIRAMISU)
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.item_quiz);
        editText1 = findViewById(R.id.editText1);
        editText2 = findViewById(R.id.editText2);
        editText3 = findViewById(R.id.editText3);
        editText4 = findViewById(R.id.editText4);
        editText5 = findViewById(R.id.editText5);
        editText6 = findViewById(R.id.editText6);
        button = findViewById(R.id.button);
    }
}
```

```

data =(mcq) getIntent().getSerializableExtra("data");
pos = getIntent().getIntExtra("pos",0);
Log.e("DATAPASSED",data.getSubject_id().toString());
editText1.setText(data.getQuestion());
editText2.setText(data.getOption_1());
editText3.setText(data.getOption_2());
editText4.setText(data.getOption_3());
editText5.setText(data.getOption_4());
editText6.setText(data.getCorrect());
button.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View view) {
        FirebaseDatabase database = FirebaseDatabase.getInstance();
        DatabaseReference myRef = database.getReference();
        if(pos==-1){
            mcq temp2 = new mcq(
                data.getSubject_id(),
                editText1.getText().toString(),
                editText2.getText().toString(),
                editText3.getText().toString(),
                editText4.getText().toString(),
                editText5.getText().toString(),
                editText6.getText().toString()
            );
            size = getIntent().getIntExtra("size",0);
            myRef.child("MCQ").child(String.valueOf(size)).setValue(temp2);
        }
        else {
            mcq temp = new mcq(
                data.getSubject_id(),
                editText1.getText().toString(),
                editText2.getText().toString(),
                editText3.getText().toString(),

```

```
        editText4.getText().toString(),
        editText5.getText().toString(),
        editText6.getText().toString()
    );
    myRef.child("MCQ").child(String.valueOf(pos)).setValue(temp);
}
});
}
}
}
```

MODEL-MCQ.JAVA

```
package com.example.newproject_testing.models;

import java.io.Serializable;

public class mcq implements Serializable {
    private Integer subject_id;
    private String question;
    private String option_1;
    private String option_2;
    private String option_3;
    private String option_4;
    private String correct;
    private final static long serialVersionUID = -450850177878496086L;
    public mcq(Integer subject_id, String question, String option_1, String option_2, String
option_3, String option_4, String correct) {
        this.subject_id = subject_id;
        this.question = question;
        this.option_1 = option_1;
        this.option_2 = option_2;
        this.option_3 = option_3;
        this.option_4 = option_4;
        this.correct = correct;
    }
    public mcq(Integer subject_id) {
        this.subject_id = subject_id;
    }
    public Integer getSubject_id() {
        return subject_id;
    }
}
```

```
public void setSubject_id(Integer subject_id) {
    this.subject_id = subject_id;
}
public String getQuestion() {
    return question;
}
public void setQuestion(String question) {
    this.question = question;
}
public String getOption_1() {
    return option_1;
}
public void setOption_1(String option_1) {
    this.option_1 = option_1;
}
public String getOption_2() {
    return option_2;
}
public void setOption_2(String option_2) {
    this.option_2 = option_2;
}
public String getOption_3() {
    return option_3;
}
public void setOption_3(String option_3) {
    this.option_3 = option_3;
}
public String getOption_4() {
    return option_4;
}
public void setOption_4(String option_4) {
    this.option_4 = option_4;
}
```

```
public String getCorrect() {  
    return correct;  
}  
public void setCorrect(String correct) {  
    this.correct = correct;  
}  
}
```

ACTIVITY_SCORE.XML

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="20dp">

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="314dp"
        android:layout_marginBottom="16dp"
        android:scaleType="centerCrop"
        android:src="@drawable/congo_images" />

    <TextView
        android:id="@+id/textViewDialogTitle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Exam Scores"
        android:textSize="18sp"
        android:textStyle="bold"
        android:gravity="center"/>

    <TextView
        android:id="@+id/textViewScores"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text=""
        android:textSize="16sp"
        android:gravity="center"/>
```

```
<Button  
    android:id="@+id/buttonDismiss"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Ok"  
    android:layout_marginTop="10dp"/>  
  
</LinearLayout>
```

SCOREACTIVITY.JAVA

```
package com.example.newproject_testing;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;

public class ScoreActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_score);
    }
}
```

ACTIVITY_NOTIFICATION.XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:orientation="vertical"
    android:padding="14dp"
    android:layout_height="match_parent">

    <androidx.recyclerview.widget.RecyclerView
        android:visibility="gone"
        android:id="@+id/recyclerView"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    </androidx.recyclerview.widget.RecyclerView>

    <LinearLayout
        android:id="@+id/linear"
        android:gravity="center"
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    </LinearLayout>

    <EditText
        android:hint="Title"
        android:id="@+id/titleET"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
    </EditText>

    <EditText
        android:hint="Message"
        android:id="@+id/messageET"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
    </EditText>

```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
</EditText>

<Button
    android:id="@+id/submit"
    android:text="Submit"
    android:layout_gravity="center"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
</Button>
</LinearLayout>
```

```
</LinearLayout>
```

ACTIVITY_NOTIFICATION2.XML

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".NotificationActivity">

</androidx.constraintlayout.widget.ConstraintLayout>
```

ITEM_NOTIFICATION.XML

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView app:contentPadding="16dp"
    android:layout_margin="12dp"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="16dp"
    android:id="@+id/card"
    xmlns:app="http://schemas.android.com/apk/res-auto">

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="12dp"
        android:layout_gravity="center"
        android:orientation="vertical">

        <TextView
            android:id="@+id/TitleTextView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="18sp"
            android:text="Title"/>

        <TextView
            android:id="@+id/DataTextView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textSize="18sp"
            android:text="Data"/>
    
```

NOTIFICATIONACTIVITY.JAVA

```
package com.example.newproject_testing;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import android.annotation.SuppressLint;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import com.example.newproject_testing.models.NotificationModel;
import com.example.newproject_testing.models.model_Subject;
import com.google.firebaseio.database.DataSnapshot;
import com.google.firebaseio.database.DatabaseError;
import com.google.firebaseio.database.DatabaseReference;
import com.google.firebaseio.database.FirebaseDatabase;
import com.google.firebaseio.database.ValueEventListener;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;
import java.util.HashMap;
import java.util.List;

public class NotificationActivity extends AppCompatActivity {
    private NotificationAdapter adapter;
    private RecyclerView recyclerView;
    private List<NotificationModel> notification;
```

```

private String course_id;
    EditText title, message;
    Button submit;
    LinearLayout linear;
    MySharedPreference preference;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_notification);
    recyclerView = findViewById(R.id.recyclerView);
    title = findViewById(R.id.titleET);
    message = findViewById(R.id.messageET);
    submit = findViewById(R.id.submit);
    linear = findViewById(R.id.linear);
    preference = new MySharedPreference(this);
    course_id = preference.getCourse_id();
    FirebaseDatabase database = FirebaseDatabase.getInstance();
    if(preference.getType_id()==0){
        recyclerView.setVisibility(View.GONE);
        linear.setVisibility(View.VISIBLE);
        Date d = Calendar.getInstance().getTime();
        submit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                DatabaseReference myRef = database.getReference();
                String tit = title.getText().toString();
                String msg = message.getText().toString();
                NotificationModel model = new NotificationModel(d.toString(),
                        tit,
                        msg,
                        course_id,
                        d.toString()
                );

```

```
        myRef.child("notification").child(d.toString()).setValue(model);
    }
});

}

else{
    if(course_id!=null) {
        DatabaseReference myRef = database.getReference("notification");
        recyclerView.setVisibility(View.VISIBLE);
        linear.setVisibility(View.GONE);
        notification = new ArrayList<>();
        myRef.addValueEventListener(new ValueEventListener() {

            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                notification.clear();
                for (DataSnapshot snapshot : dataSnapshot.getChildren()) {
                    NotificationModel model = snapshot.getValue(NotificationModel.class);
                    if (model != null && course_id.equals(model.getCourse_id())) {
                        notification.add(model);
                    }
                }
                recyclerView.setLayoutManager(new
                LinearLayoutManager(NotificationActivity.this));
                adapter = new NotificationAdapter(notification, NotificationActivity.this);
                recyclerView.setAdapter(adapter);
            }
        });
    }
    @Override
    public void onCancelled(@NonNull DatabaseError error) {
        Log.e("TestingFirebase", "Failed to read value.", error.toException());
    }
});
}
}
}
```

NOTIFICATIONADAPTER.JAVA

```
package com.example.newproject_testing;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;
import com.example.newproject_testing.models.NotificationModel;
import java.util.List;

public class NotificationAdapter extends
RecyclerView.Adapter<NotificationAdapter.ViewHolder> {
    List<NotificationModel> modelList;
    Context context;
    public NotificationAdapter(List<NotificationModel> modelList, Context context) {
        this.modelList = modelList;
        this.context = context;
    }
    @NonNull
    @Override
    public NotificationAdapter.ViewHolder onCreateViewHolder(@NonNull ViewGroup parent,
    int viewType) {
        View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.item_notification,
        parent, false);
        return new NotificationAdapter.ViewHolder(view);
    }
    @Override
    public void onBindViewHolder(@NonNull NotificationAdapter.ViewHolder holder, int
    position) {
```

```
holder.TitleTextView.setText(modelList.get(position).getTitle());
holder.DataTextView.setText(modelList.get(position).getMessage());
}

@Override
public int getItemCount() {
    return modelList.size();
}
public class ViewHolder extends RecyclerView.ViewHolder {
    TextView TitleTextView,DataTextView;
    public ViewHolder(@NonNull View itemView) {
        super(itemView);
        DataTextView = itemView.findViewById(R.id.DataTextView);
        TitleTextView = itemView.findViewById(R.id.TitleTextView);
    }
}
```

CHAPTER 8:

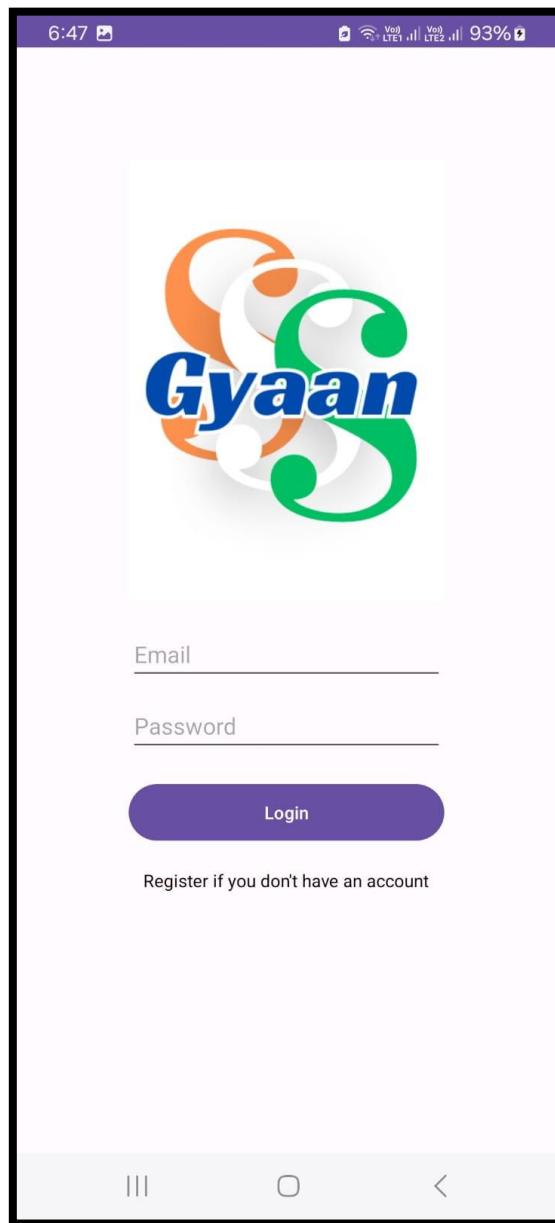
FORM DESIGN

CHAPTER 8: FORM DESIGN

SPLASH SCREEN



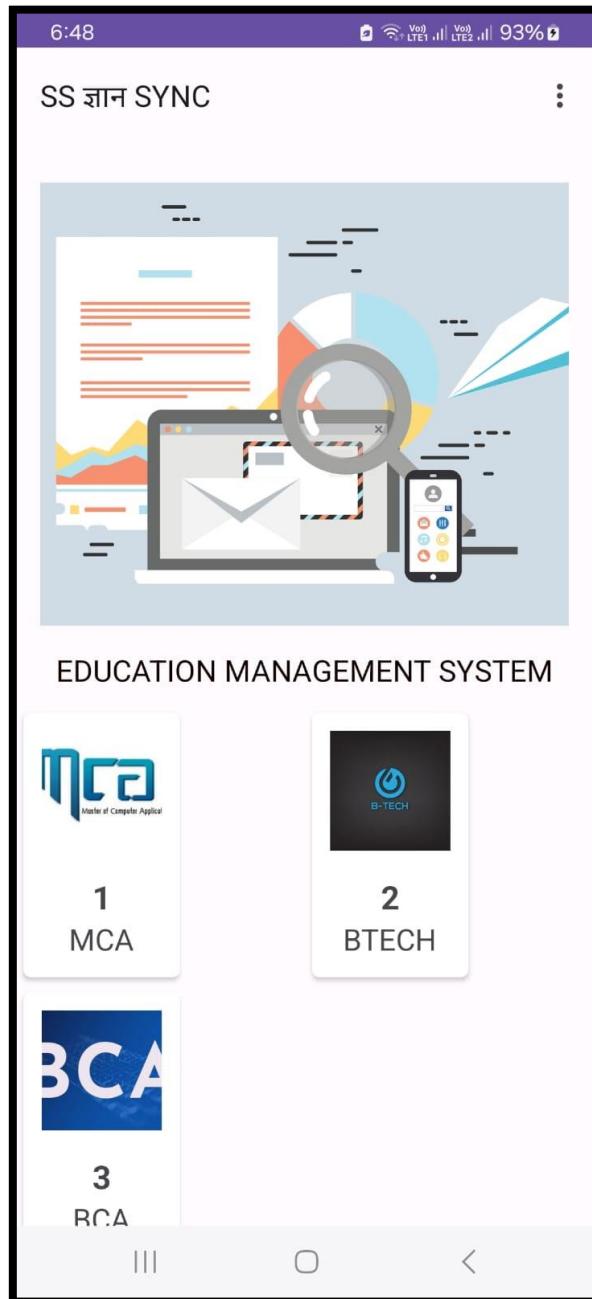
LOGIN SCREEN



STUDENT LOGON



MAIN SCREEN



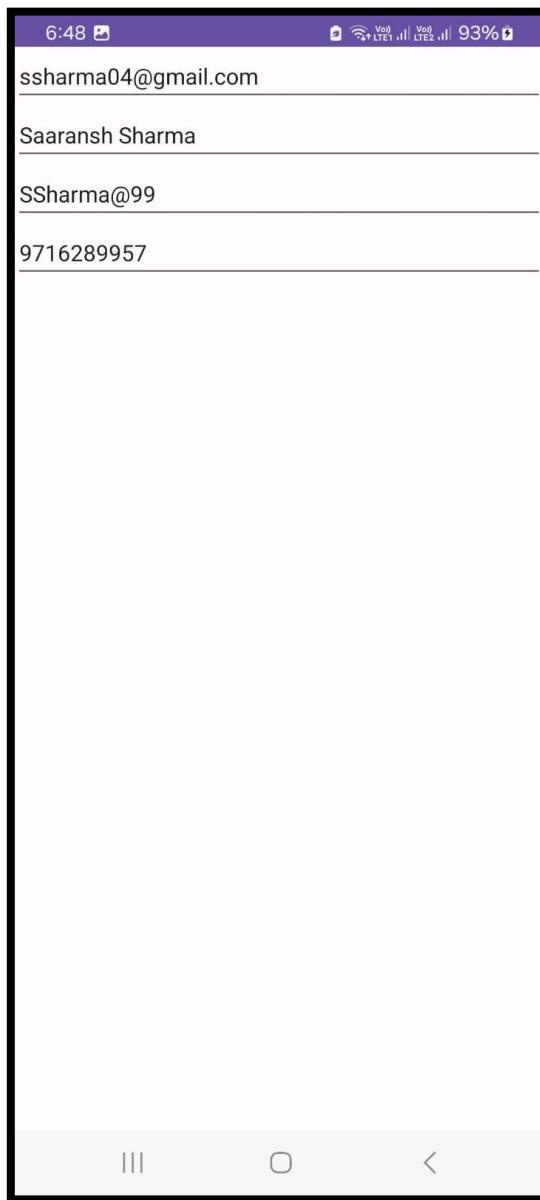
MENU BAR SCREEN



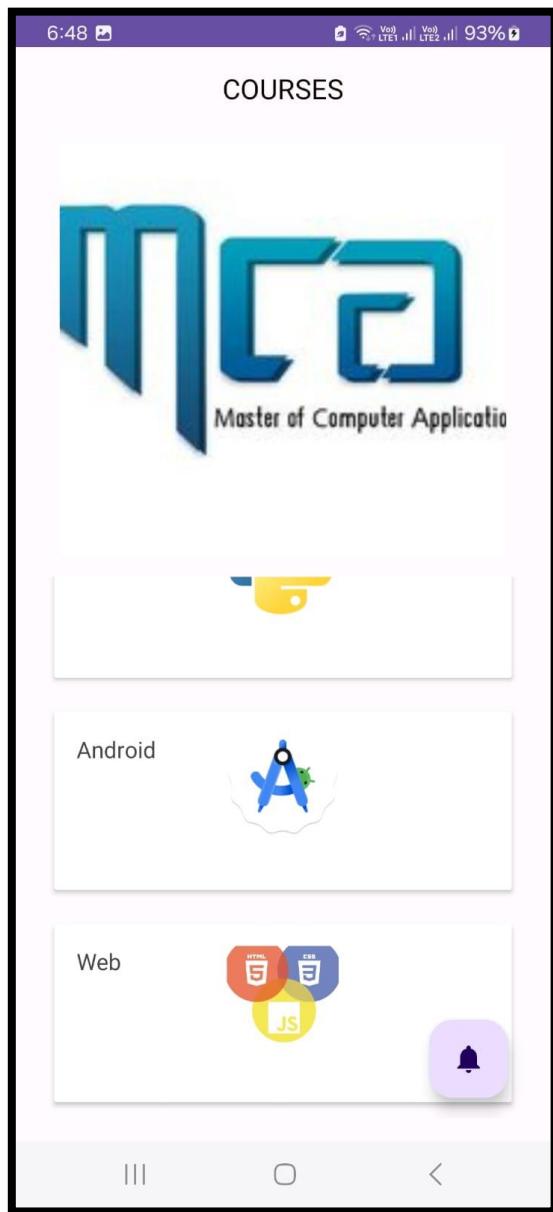
ABOUT SCREEN



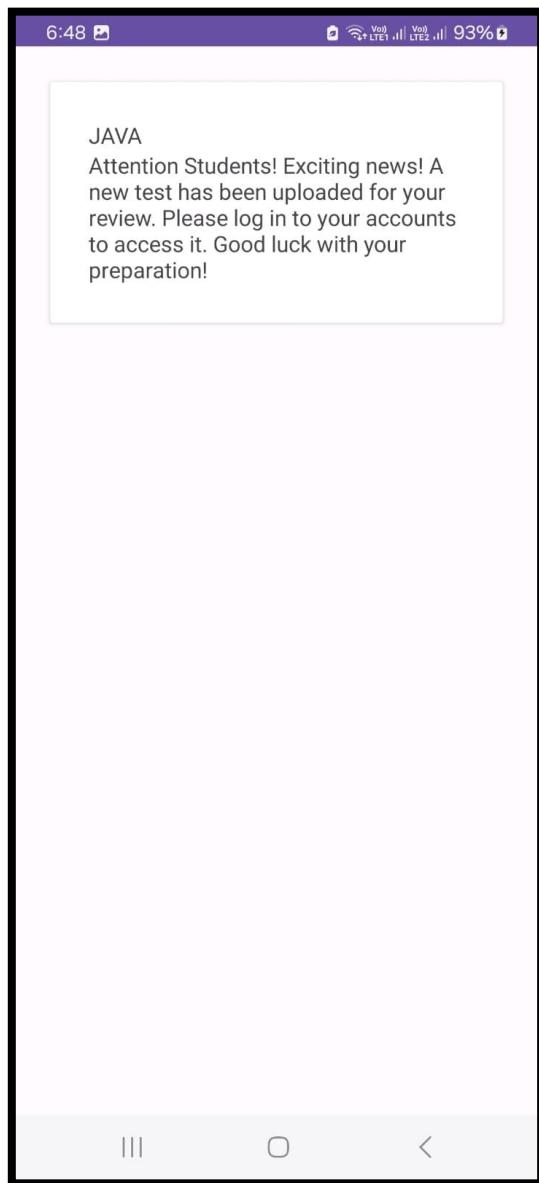
PROFILE SCREEN



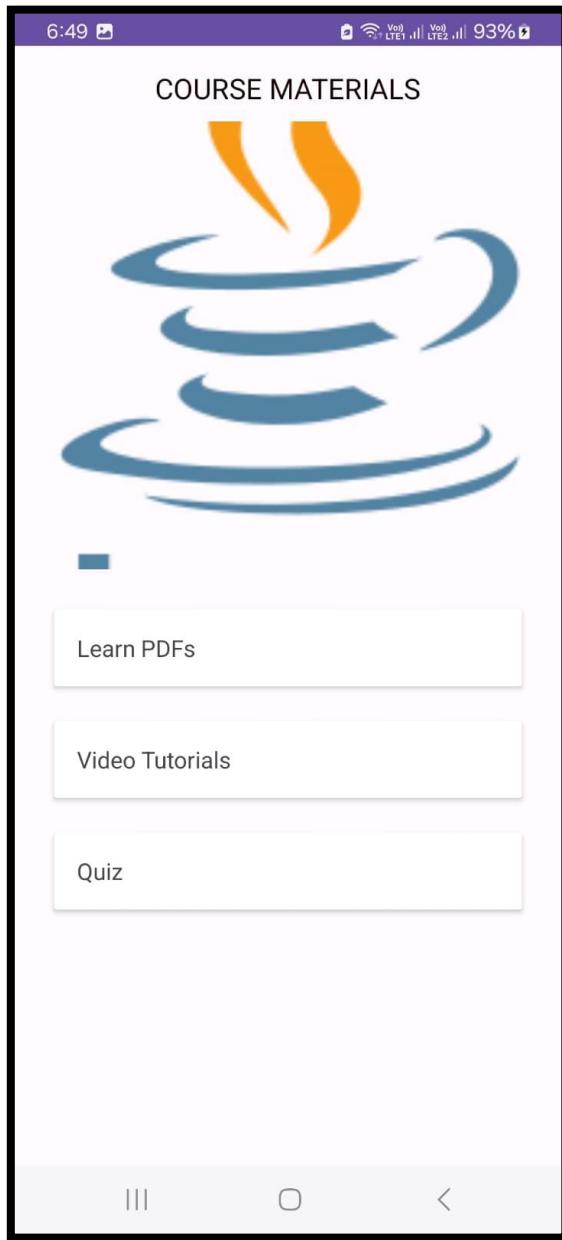
COURSE SCREEN



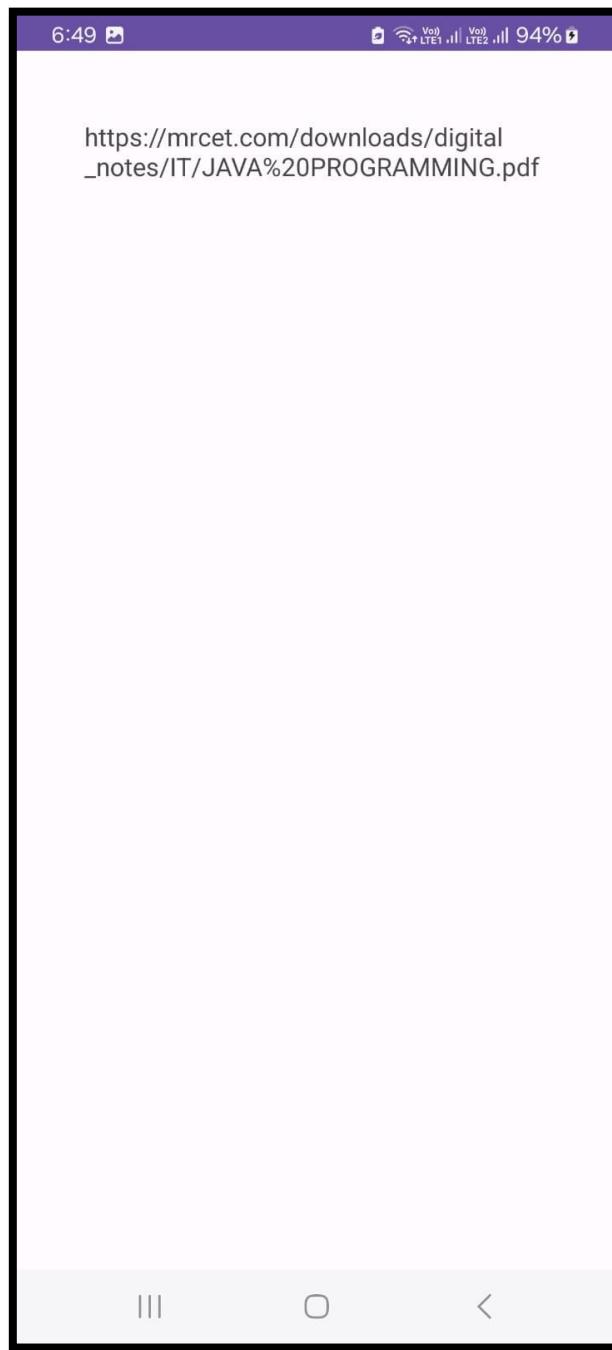
NOTIFICATION SCREEN



SUBJECT SCREEN



PDF SCREEN



**DIGITAL NOTES
ON
JAVA PROGRAMMING (R20A0508)**

**B.TECH II YEAR - II SEM
(2021-22)**



DEPARTMENT OF INFORMATION TECHNOLOGY

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY
(Autonomous Institution – UGC, Govt. of India)
(Affiliated to JNTUH, Hyderabad, Approved by AICTE - Accredited by NBA & NAAC - 'A' Grade - ISO 9001:2015 Certified)
Maisammaguda, Dhulapally (Post Via. Hakimpet), Secunderabad – 500100, Telangana State, INDIA.

[JAVA PROGRAMMING] Page 1

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY
DEPARTMENT OF INFORMATION TECHNOLOGY

II Year B.Tech IT – II Sem

L T P/D C
4 1/- 4

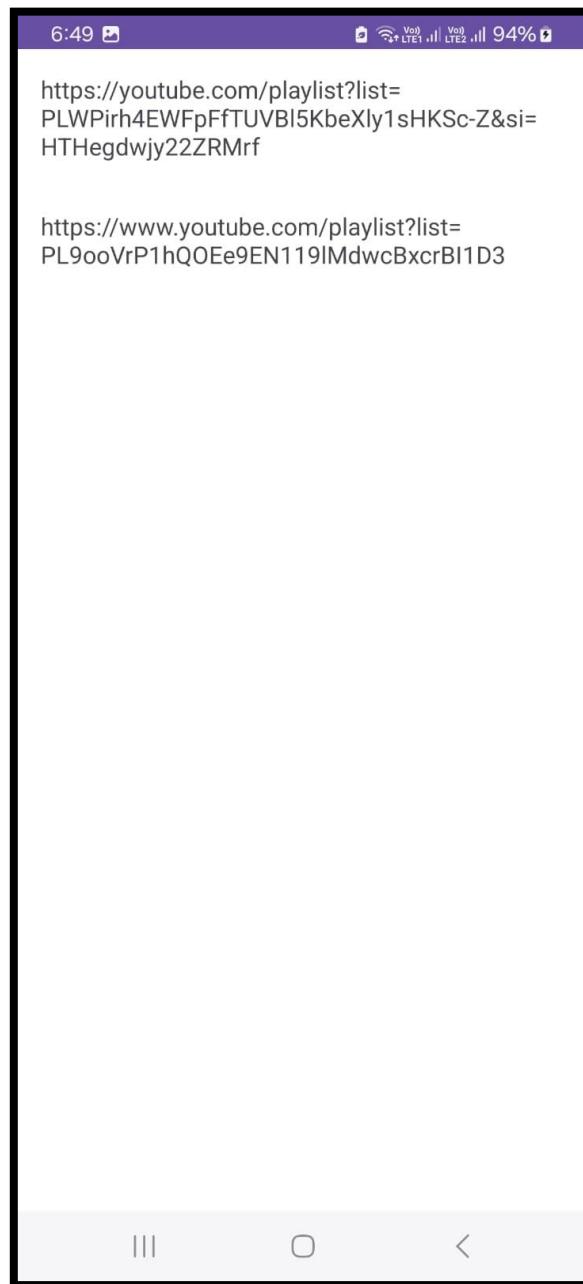
Activities: [RIBA0509]JAVA PROGRAMMING

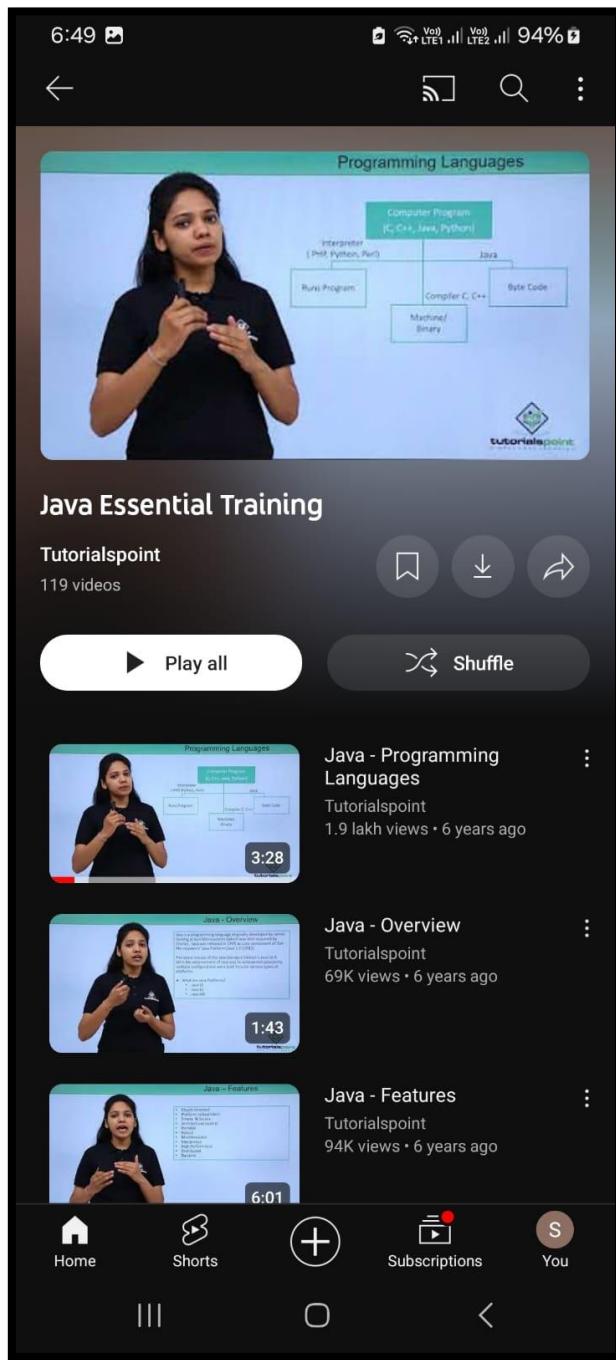
1 / 118

Download finished.

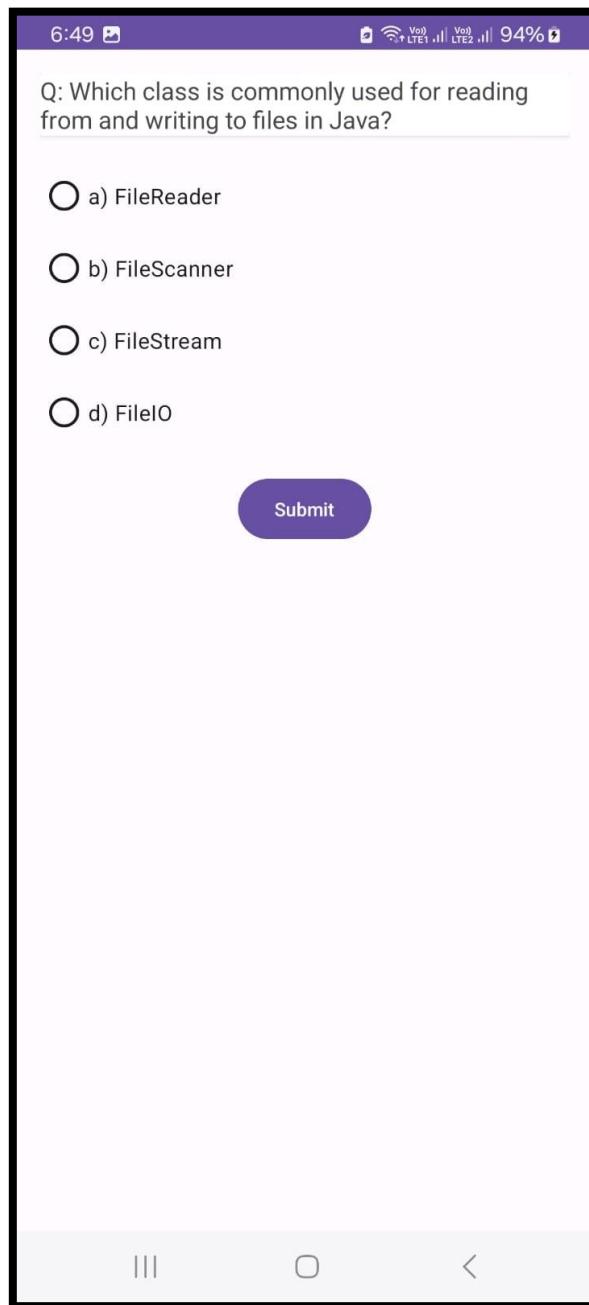


VIDEO SCREEN

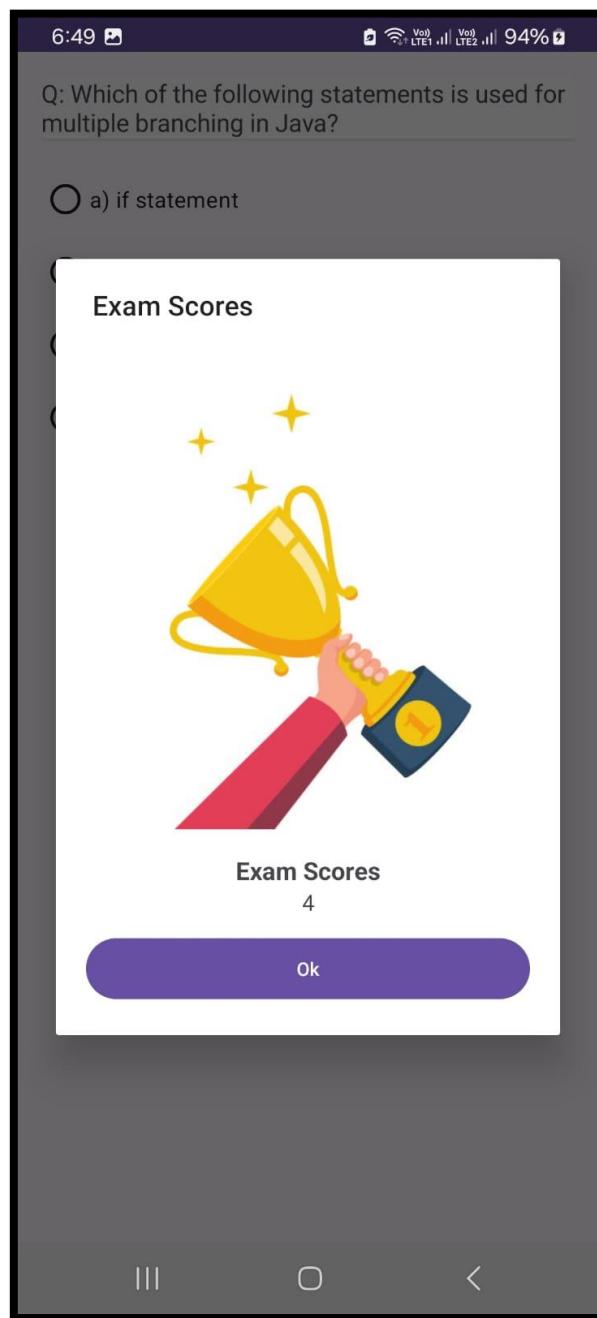




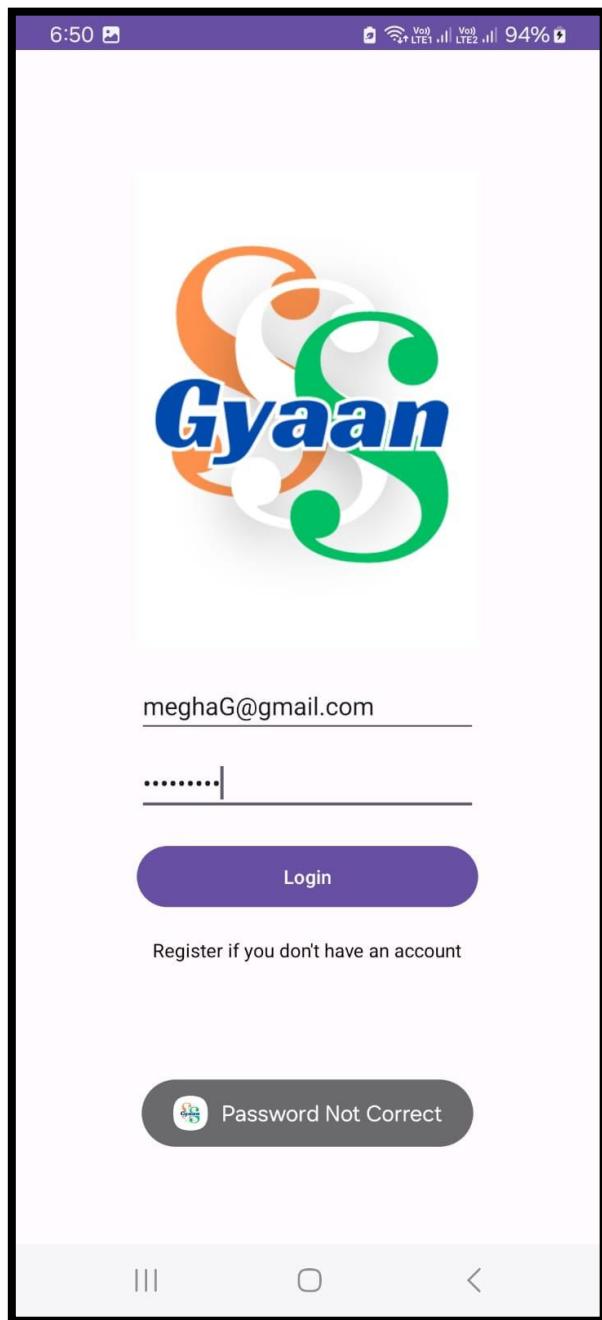
QUIZ SCREEN



SCORE SCREEN



TEACHER LOGIN



MAIN SCREEN



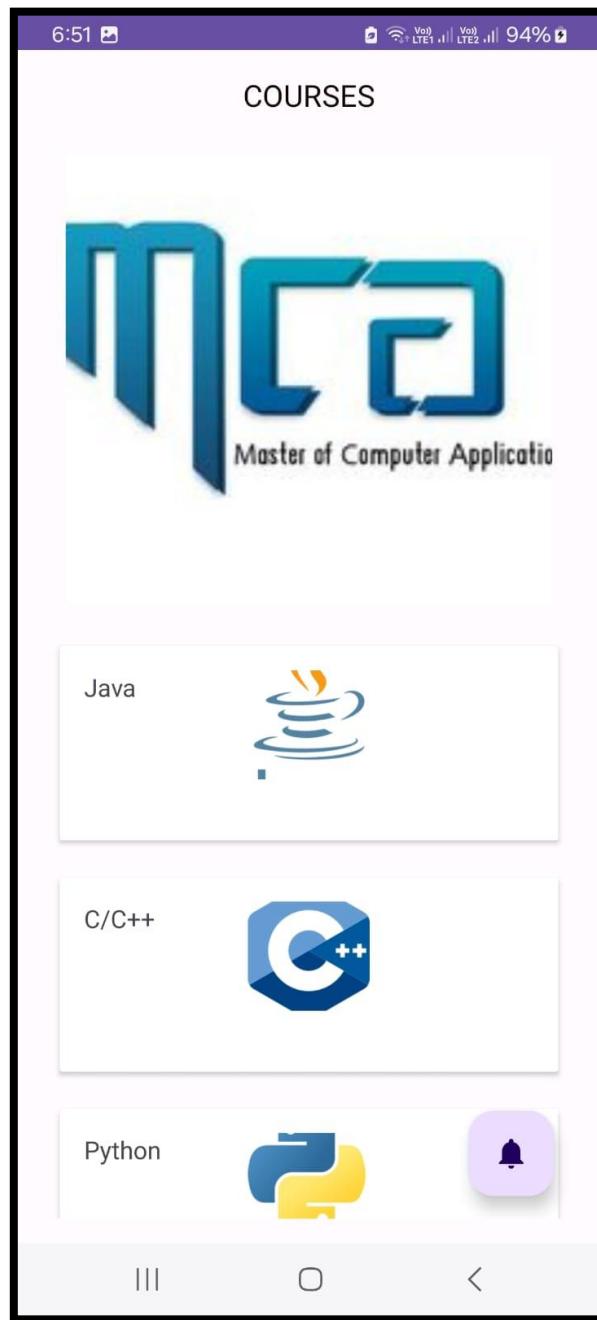
ABOUT SCREEN



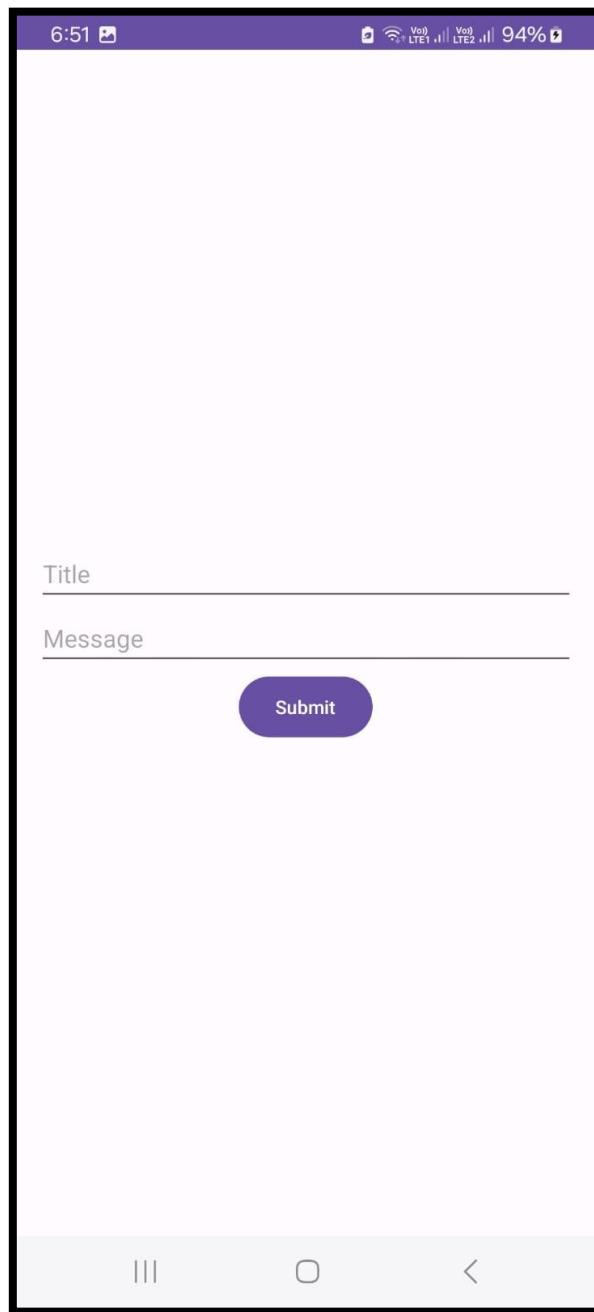
PROFILE SCREEN (TEACHER)



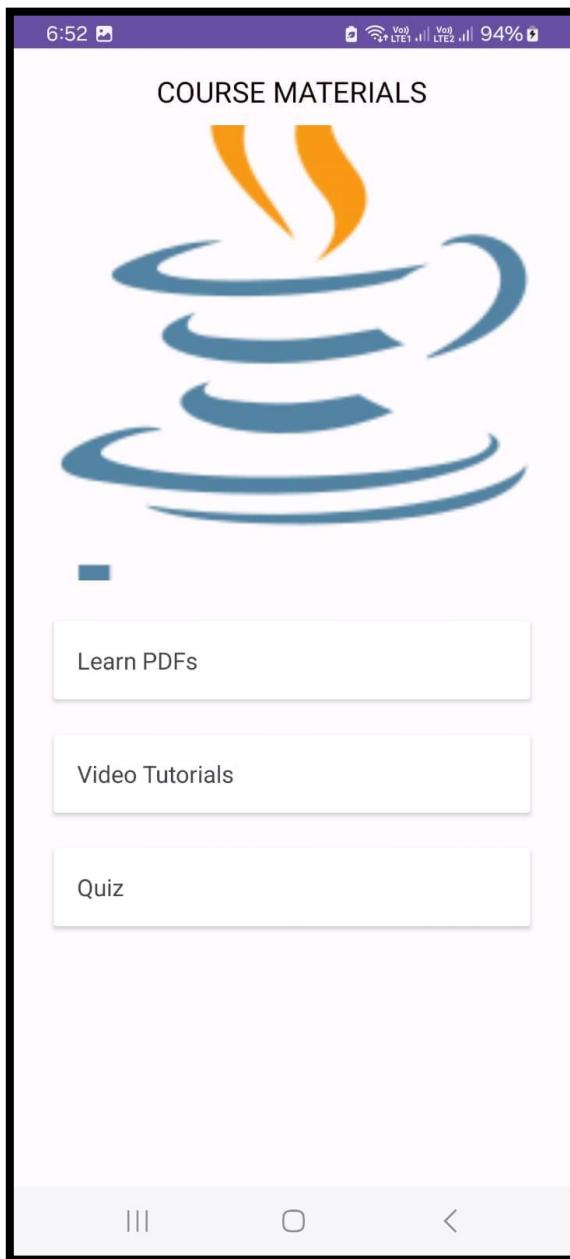
COURSE SCNREE



NOTIFICATION SCREEN



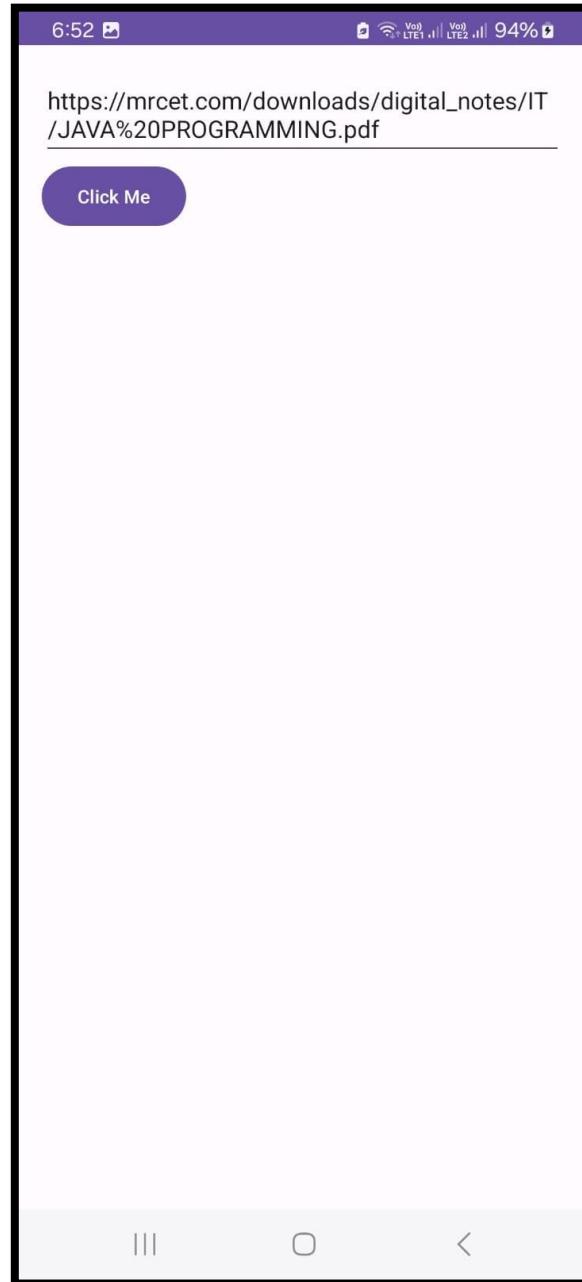
SUBJECT SCREEN



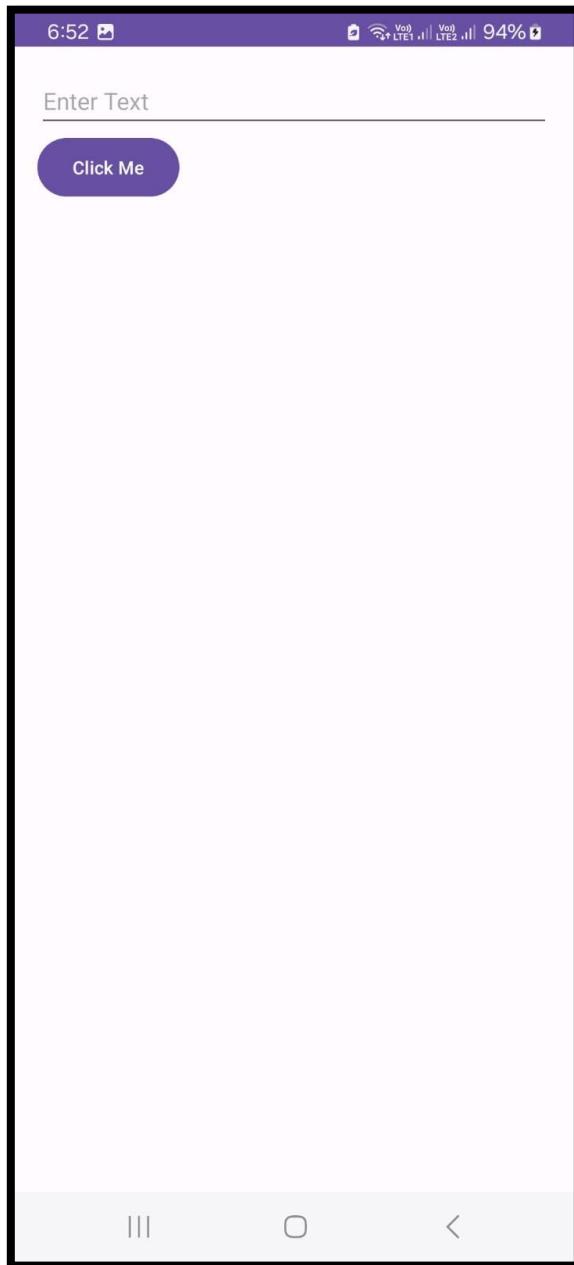
PDF SCREEN



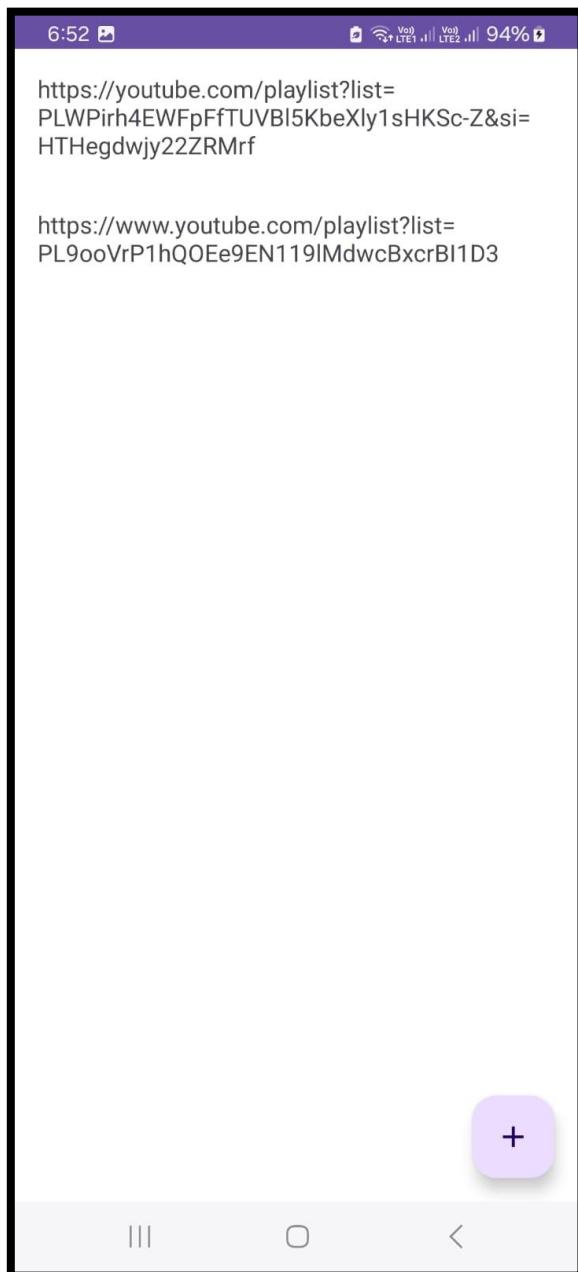
PDF EDIT SCREEN



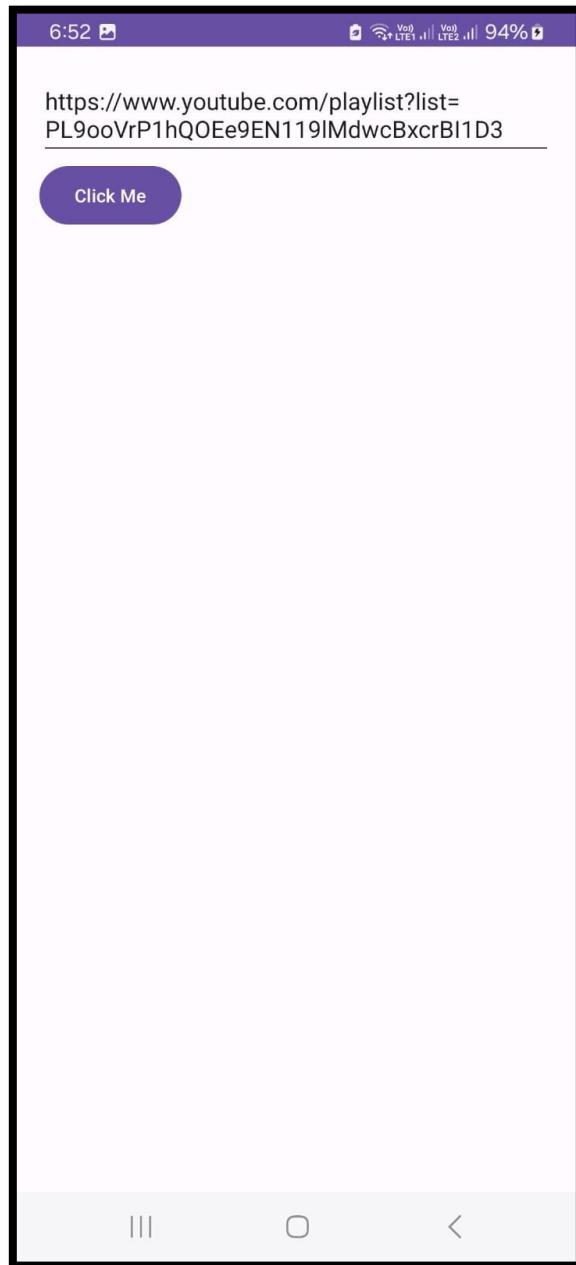
PDF ADDING SCREEN



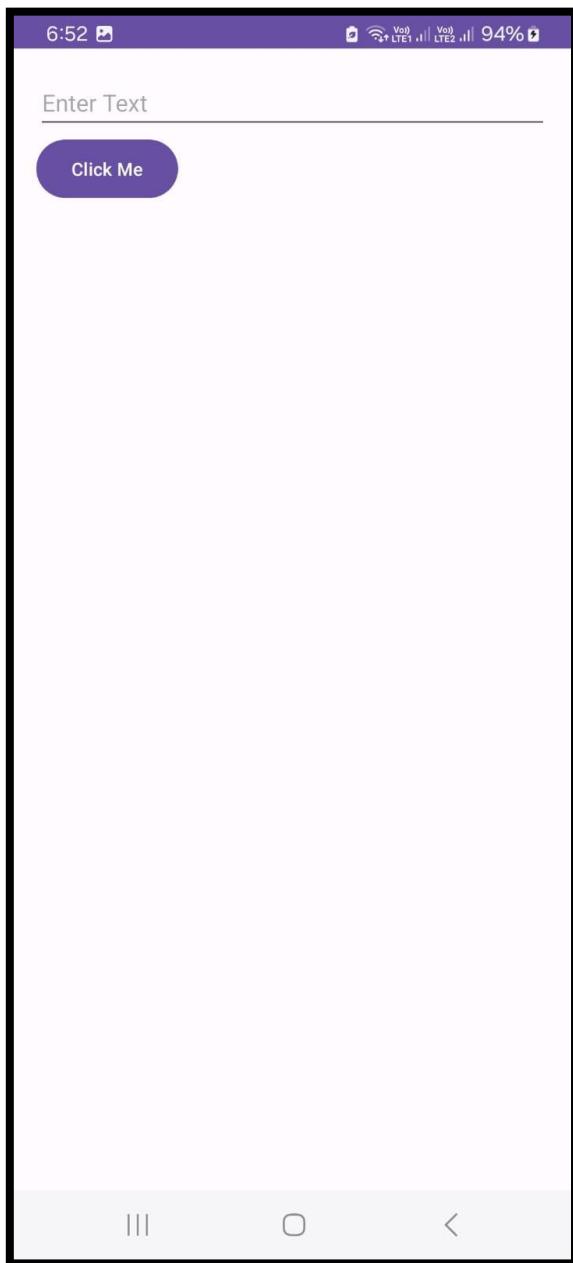
VIDEO SCREEN



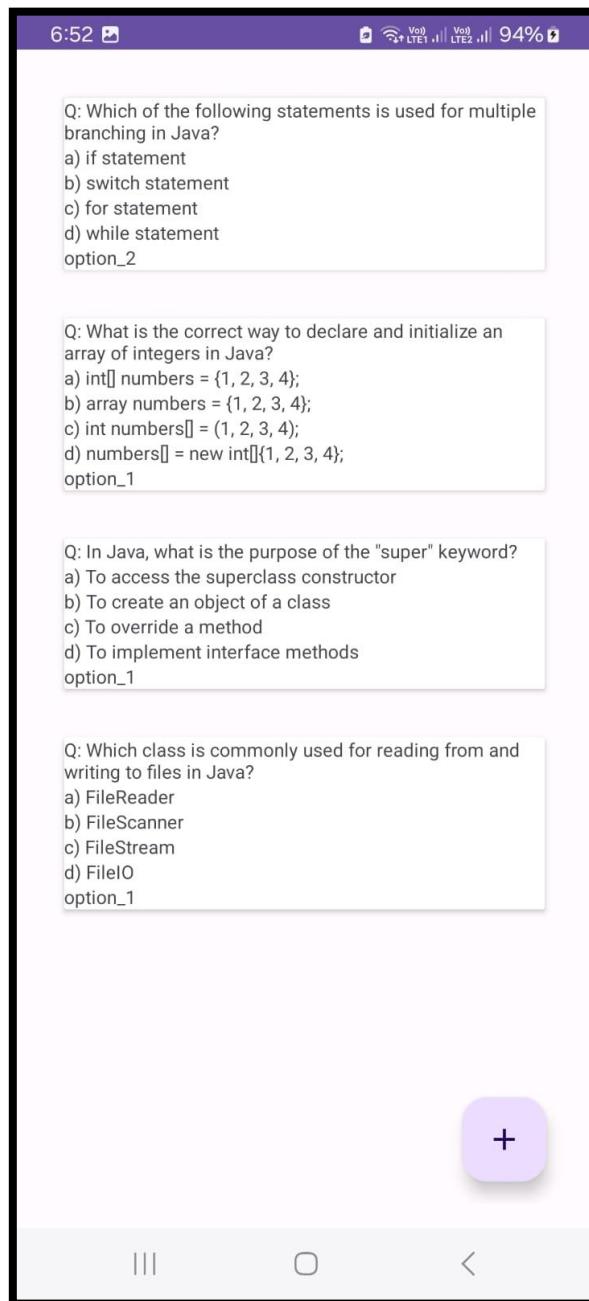
VIDEO EDITING SCREEN



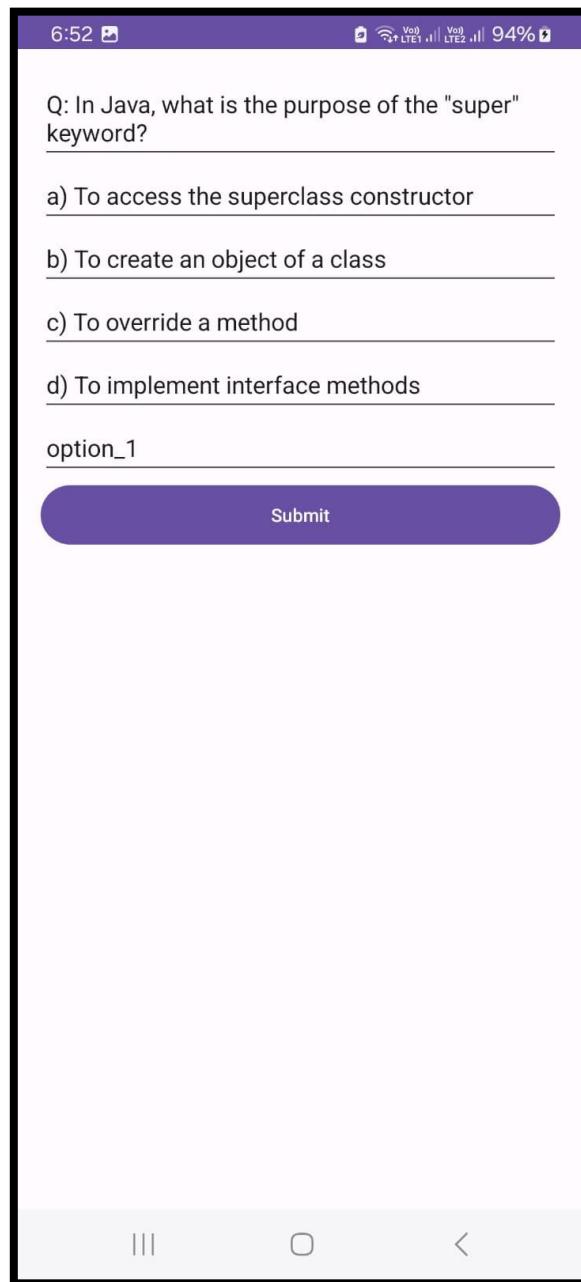
VIDEO ADDING SCREEN



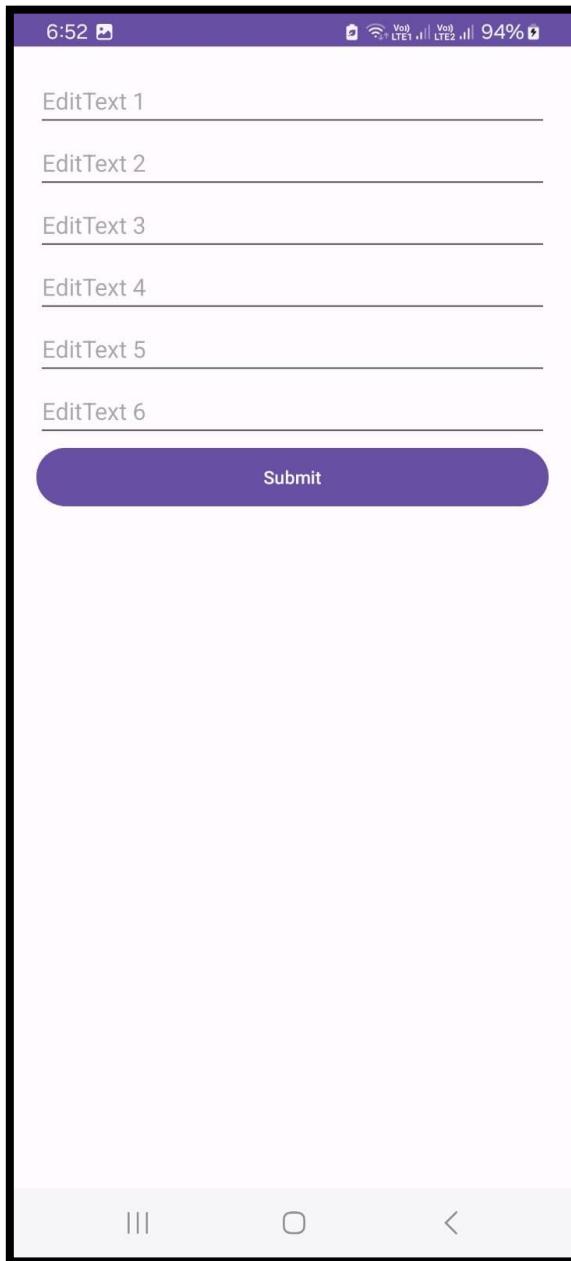
QUIZ SCREEN



QUIZ EDITING SCREEN



QUIZ ADDING SCREEN



CHAPTER 9:

TESTING

MECHANISMS

USED

CHAPTER 9: TESTING MECHANISMS USED

9.1. EMULATOR TESTING

SS GYAN SYNC HUB undergoes rigorous testing the usage of emulators to make certain compatibility across exclusive gadgets and display screen sizes. Emulator trying out lets in developers to simulate diverse tool configurations and working system variations, facilitating comprehensive trying out without physical devices. By running the software on emulated environments, capacity performance, responsiveness, and person revel in problems are recognized and addressed early within the improvement method. Emulator trying out enables ensure that SS GYAN SYNC HUB grants a steady and reliable revel in to customers, regardless of their device.

9.2. MANUAL TESTING

Manual testing performs a crucial position in validating the functionality and value of SS GYAN SYNC HUB. A dedicated team of testers systematically evaluates each characteristic and module of the platform, following predefined take a look at instances and scenarios. Manual trying out encompasses various aspects, including user authentication, navigation, content material get entry to, shape submissions, and notification handling. Testers confirm that the platform meets distinct necessities, adheres to design recommendations, and grants an intuitive consumer revel in. Additionally, guide testing lets in testers to discover any usability troubles, inconsistencies, or facet instances that may not be captured through automatic checking out. Through meticulous manual trying out, SS GYAN SYNC HUB guarantees excessive first-rate, reliability, and person pleasure.

By employing both emulator checking out and guide checking out methodologies, SS GYAN SYNC HUB undergoes thorough evaluation to guarantee its functionality, performance, and usefulness. These trying out mechanisms paintings in tandem to perceive and remedy problems, ensuring that the platform meets the best standards of first-class and supplies a continuing educational revel in for educators and learners alike.

CHAPTER 10:

IMPLEMENTATION

MECHANISMS

CHAPTER 10: IMPLEMENTATION MECHANISMS

The implementation of SS GYAN SYNC HUB revolves around a streamlined project flow designed to provide users with intuitive navigation and seamless access to educational resources.

The following steps outline the project flow for both teachers and students:

- **Login/Register:**
 - Users first access the platform and are prompted to either log in with their existing credentials or register for a new account.
 - The Sign-In Form module ensures secure user authentication, requiring users to input their email and password.
 - For new users, the Registration Form module provides a user-friendly interface to create an account, capturing essential information such as email, name, phone number, and password. Radio buttons allow users to specify their role as either a teacher or a student.
- **Course Selection:**
 - Upon successful login, users are directed to the Course module, which presents a categorized list of educational programs available on the platform.
 - Users can easily navigate through the list of courses, selecting their desired program from options like B.Sc, B.C.A., M.C.A., B.Tech, M.Tech, and more.
- **Subject Navigation:**
 - After selecting a course, users are taken to the Subject module, which organizes subjects related to the chosen program.
 - Subjects such as Java, C/C++, Python, Android Development, Web Development, and various engineering and academic subjects are presented in a user-friendly format.
- **Task Selection:**
 - Within each subject, users can choose from different tasks such as accessing PDF resources, watching videos, or attempting quizzes.

- For teachers, the Teacher module provides additional functionalities to manage and create educational resources, including posting PDFs, managing videos, and creating quizzes.
 - For students, the Student module allows access to PDFs, videos, and quizzes for learning and assessment purposes.
-
- **Task Completion:**
 - Users can seamlessly complete their chosen tasks, whether it's viewing educational materials, watching videos, or attempting quizzes.
 - Progress tracking and feedback mechanisms ensure that users can monitor their learning journey and review their performance.
 - **Notification Management:**
 - The Notification module plays a crucial role in keeping users informed about important updates and announcements within the platform.
 - Users receive real-time notifications regarding new resources, upcoming quizzes, deadlines, and other relevant information, ensuring timely communication.

CHAPTER 11: CONCLUSIONS

CHAPTER 11: CONCLUSIONS

The SS GYAN SYNC HUB schooling management system software offers a complete platform designed to streamline educational methods, facilitate efficient conversation between instructors and students, and decorate the general gaining knowledge of revel in. Throughout the development and evaluation phases, numerous findings, barriers, and areas for possibilities had been recognized.

11.1. FINDINGS

- ***Efficient User Management:*** The machine affords an intuitive interface for each teachers and students, permitting seamless account introduction and access to instructional assets. The Sign-In and Registration Form modules make sure stable consumer authentication, at the same time as the Course and Subject modules enable easy navigation and choice of relevant academic packages and subjects.
- ***Effective Content Management:*** The Teacher module empowers educators to manipulate educational sources efficiently via the PDF Link, Video Link, and Quiz submodules. Teachers can add, edit, and delete PDF documents and video resources, in addition to create and administer quizzes to assess students' expertise.
- ***Comprehensive Learning Tools:*** The Student module equips inexperienced persons with the necessary equipment to get right of entry to and interact with academic content material. From viewing PDF sources and videos to attempting quizzes, college students can tailor their getting to know experience to suit their options and educational desires.
- ***Timely Notifications:*** The Notification module ensures timely dissemination of updates and announcements, fostering a dynamic and knowledgeable gaining knowledge of surroundings. Users acquire notifications approximately course updates, new sources, and critical announcements, preserving them engaged and knowledgeable.

11.2. LIMITATIONS

- ***Limited Platform Accessibility:*** Currently, the utility is developed for Android devices simplest, limiting its accessibility to customers who may also choose other systems which includes iOS or internet-based interfaces. Extending aid to multiple systems may want to beautify the attain and usefulness of the system.
- ***Dependency on Internet Connectivity:*** The machine relies on Internet connectivity for accessing and updating information stored on the Firebase backend. Users in areas with poor

net connectivity might also face challenges in having access to educational resources and participating in interactive activities inclusive of quizzes.

- **Scalability Concerns:** While the present day system structure supports a significant number of users, scalability issues may get up as the consumer base grows. Adequate measures want to be carried out to make sure the device can accommodate growing user needs without compromising overall performance and reliability.

11.3. SCOPE FOR FUTURE PROSPECTS

- **Cross-Platform Compatibility:** Expanding the utility to assist more than one structures, such as iOS and net-primarily based interfaces, would develop its accessibility and cater to a extra various consumer base.
- **Offline Access:** Implementing offline get right of entry to functions could permit users to download educational resources for offline viewing, permitting uninterrupted studying even in regions with constrained net connectivity.
- **Advanced Analytics:** Incorporating analytics gear to track person engagement, performance metrics, and gaining knowledge of consequences may want to offer treasured insights for educators and directors to optimize coaching techniques and route content.
- **Enhanced Collaboration Tools:** Introducing collaborative functions including dialogue boards, group tasks, and real-time communication channels would foster interactive learning environments and facilitate peer-to-peer information sharing.
- **Integration with Learning Management Systems:** Integrating the software with current mastering management systems (LMS) utilized by educational institutions could streamline administrative procedures and offer a unbroken enjoy for each teachers and college students.

CHAPTER 12:

BIBLIOGRAPHY

CHAPTER 12: BIBLIOGRAPHY

- **Books:**
 - *Tanenbaum, A. S., & Bos, H. (2015). Modern Operating Systems (4th ed.). Pearson.*
 - *This book provides insights into operating systems, which may help understand the underlying principles of system development.*
- **Articles:**
 - *Smith, J. (2019). "Mobile Application Development: Best Practices and Considerations." Journal of Mobile Technology, 8(2), 45-58.*
 - *This article discusses best practices in mobile application development, which could inform your approach to building the front end of your education management system.*
 - *Johnson, R., & Smith, A. (2020). "Firebase: A Comprehensive Guide to Backend as a Service (BaaS) Development." Journal of Cloud Computing, 15(3), 112-125.*
 - *This article offers insights into utilizing Firebase as a backend service, which is relevant to your system's backend implementation.*
- **Online Resources:**
 - *Android Developers. (n.d.). Android Studio Documentation. Retrieved from <https://developer.android.com/studio>*
 - *The official documentation for Android Studio provides comprehensive guidance on building Android applications, including UI design and coding practices.*
 - *Firebase Documentation. (n.d.). Retrieved from <https://firebase.google.com/docs>*
 - *The Firebase documentation offers detailed instructions and examples for integrating Firebase services into your application's backend.*
- **Frameworks and Libraries:**
 - *Material Design Components for Android. (n.d.). Retrieved from <https://material.io/develop/android/docs/getting-started>*
 - *Material Design Components provide guidelines and components for creating visually appealing and consistent user interfaces in Android applications, which aligns with your front-end development in Android Studio.*

- **Websites:**
 - *Stack Overflow.* (n.d.). <https://stackoverflow.com/>
 - *Stack Overflow is a valuable resource for troubleshooting coding issues and seeking advice from the developer community during the development process.*
- **Tutorials:**
 - *Android Studio Tutorial for Beginners.* (n.d.). Retrieved from <https://www.youtube.com/watch?v=BGCBb8HPrOo>
 - *This tutorial series on YouTube provides step-by-step guidance for beginners in Android Studio, which can be beneficial for learning essential concepts and techniques.*
 - *Firebase Tutorial for Android App Development.* (n.d.). Retrieved from <https://www.youtube.com/watch?v=9kRgVxULbag>
 - *This tutorial covers Firebase integration in Android applications, offering practical demonstrations and explanations that can aid in implementing the backend functionalities of your system.*

**APPENDICES
&
PLAGIARISM CHECKER**

APPENDICES

(a) Software Requirement Specifications (SRS)

1. Introduction

The SS Gyan Sync Hub is an Education Management System application designed to facilitate seamless interaction between instructors and students, providing a platform for instructional content delivery, checks, and communication. This report outlines the practical and non-practical requirements of the gadget to make sure its effective development and deployment.

2. Functional Requirements

2.1 User Authentication

- The device shall provide a Sign-In Form for customers to soundly access the platform.
- Users shall authenticate using their email and password.
- New customers will be able to sign in thru a Registration Form, offering important details inclusive of name, e-mail, telephone, and password.

2.2 Course Management

- The device shall categorize courses such as B.Sc, B.C.A., M.C.A., B.Tech, M.Tech, etc.
- Each path shall have a completely unique identifier, course call, and an associated icon for visual identity.

2.3 Subject Management

- Subjects shall be organized below respective guides.
- Each subject shall have a unique identifier, situation call, associated direction ID, and an icon for visible illustration.

2.4. Resource Management (Teachers)

- Teachers shall be capable of upload, delete, and edit PDF resources for subjects they're assigned to.
- Teachers will be able to control video assets associated with unique topics.
- Teachers shall have the capacity to create and manipulate quizzes for assessing college students.

2.5 Resource Access (Students)

- Students shall be able to get admission to PDF resources uploaded by teachers for their respective subjects.
- Students shall have get entry to to video resources associated with their subjects for more desirable getting to know.
- Students shall be capable of attempt quizzes created by way of instructors and overview their consequences.

2.6 Notification System

- The machine shall offer a notification function to disseminate essential updates and announcements.
- Notifications will be dispatched to users to preserve them informed approximately platform-associated activities and events.

3. Non-Functional Requirements

3.1 Security

- User authentication will be stable to save you unauthorized get right of entry to.
- Data transmission and garage shall adhere to industry-general encryption protocols to shield consumer statistics.

3.2 Performance

- The device will be responsive, supplying quick get right of entry to to sources and functionalities.
- It shall be capable of managing concurrent user requests effectively without great overall performance degradation.

3.3 Scalability

- The machine architecture shall be designed to deal with destiny scalability requirements, taking into consideration the addition of new guides, subjects, and users with out impacting performance.

3.4 Reliability

- The device shall ensure high availability, minimizing downtime for users.
- It shall have mechanisms in vicinity to recover gracefully from failures to maintain continuity of provider.

3.5 User Experience

- The consumer interface shall be intuitive and consumer-pleasant, making sure ease of navigation for each teacher and students.
- The device shall assist multi-platform get right of entry to, together with web browsers and cell programs, to cater to numerous person choices.

4. Database Design

The system shall make use of Firebase because the backend database, dependent in step with the following schema:

- Users: Separate collections for instructors and students, storing person statistics consisting of e mail, call, telephone, and password.
- Course: Stores information approximately different courses which include course ID, name, and icon.
- Subject: Contains info of topics along with challenge ID, name, related direction ID, and icon.
- Data_PDF: Stores PDF resources with links and associated challenge IDs.
- Data_VIDEO: Stores video sources with links and related problem IDs.
- MCQ: Stores quiz questions, options, accurate solutions, and associated subject IDs.

5. Conclusion

The SS Gyan Sync Hub Education Management System goals to offer a complete platform for academic content material transport and assessment. By adhering to the outlined useful and non-practical necessities, the system will make certain a seamless person enjoy at the same time as preserving safety, reliability, and scalability.

(b) User Manual: Interactive Learning Platform

Welcome to our interactive learning platform! This user guide will manual you via the steps of the usage of our utility, whether or not you are a student keen to research or a instructor equipped to share information.

1. Opening the Application:

When you open the software, you'll be greeted with a login display.

2. Login Process:

- a. Enter the credentials (username and password) that you've selected throughout registration.
- b. The device will test in case you're a registered person and if the supplied credentials are correct.
- c. If the login is a hit, you will be directed to the primary interface. If now not, follow the steps for the registration procedure.

3. Registration Process:

- a. If you are a new consumer, click on the registration button.
- b. Provide the necessary information including your name, electronic mail, and password.
- c. Choose whether you are registering as a teacher or a pupil.
- d. Submit the registration form.
- e. You'll be redirected lower back to the login page. Now, proceed with the login technique.

4. Main Interface:

Upon successful login, you may be directed to the primary interface.

5. Actions for Teachers:

a. If you're logged in as a teacher:

- i. You'll see alternatives to feature notifications, PDF links, video playlist hyperlinks, and quizzes.

- ii. You can upload/edit notifications to speak essential information on your college students.
- iii. Add/edit PDF hyperlinks to provide extra reading substances.
- iv. Add/edit video playlist hyperlinks for supplementary learning materials.
- v. Add/edit quizzes to evaluate your students' information.

6. Actions for Students:

a. If you are logged in as a pupil:

- i. You'll see notifications furnished via your teachers.
- ii. Access PDF hyperlinks for extra study materials.
- iii. Watch video playlists shared by using your instructors for stronger getting to know.
- iv. Attempt quizzes supplied with the aid of your instructors and receive on the spot outcomes.

7. Logout Option:

- a. At any factor, if you wish to log off, definitely click on the logout button.
- b. You'll be securely logged out of the machine and redirected back to the login page.

In Summary:

Our interactive studying platform offers a seamless revel in for both teachers and students. Teachers can without difficulty control path substances and checks, at the same time as college students have get admission to to a variety of assets to enhance their studying adventure. With consumer-pleasant interfaces and sturdy features, our platform objectives to facilitate a dynamic and attractive instructional enjoy for all customers.

Start exploring and enriching your mastering experience nowadays!!!

PLAGIARISM CHECKER

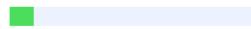
The screenshot shows the main interface of the Plagiarism Checker X software. At the top, there's a blue header bar with the text "New Update is Available 9.0.0" and a "Update Now" button. Below the header, the word "Summary" is prominently displayed in bold. A sub-header below it says "Scanned content overall summary report." To the right of the summary text is a "View Report" button. On the left side of the main content area, there's a sidebar with various navigation options: "Dashboard", "Online Plagiarism", "Side By Side Difference", "Bulk Comparison", and "Settings". Below these options, there's a status message "Activated (Enterprise) v8.0.7" and a "Update Available" button. The central part of the screen features a large pie chart titled "Share Score with Friends". The chart is divided into two segments: a dark blue segment representing "Plagiarized (10%)" and a bright green segment representing "Unique (90%)". Below the pie chart, there's a table with three rows: "Date" (Apr 22, 2024), "Words" (1202 Words Plagiarized / Total Words 12415), and "Source" (31 Source(s) Identified). The entire interface has a clean, modern design with a light gray background.



Plagiarism Checker X - Report

Originality Assessment

10%



Overall Similarity

Date: Apr 22, 2024

Matches: 1202 / 12415 words

Sources: 31

Remarks: Low similarity

detected, check with your supervisor if changes are required.

Verify Report:

V rfc t nl n

Interface: a. Upon successful login, redirect to the main interface. b. Display different courses available. 4. Actions for Teacher: a. If 14 logged in as a teacher: i. Display options to add notifications, PDF links, video playlist links, and quizzes. ii. Allow the teacher to add/edit notifications. iii. Allow the teacher to add/edit pdf links. iv. Allow the teacher to add/edit video playlist links. v. Allow the teacher to add/edit quizzes. 5. Actions for Student: a. If logged in as a student: i. Display notifications. ii. Display PDF links. iii. Display video playlist links. iv. Allow students to attempt quizzes and receive results. 6. Logout Option: a. Provide a logout option for both teachers and students. b. Upon logout, redirect back to the login page.

first identify the functional dependencies and then create tables based on those dependencies. Functional Dependencies:

- 24 First Normal Form (1NF):
 - o All attributes must have atomic values.
 - o There should be no repeating groups or arrays.
- Second Normal Form (2NF):
 - o Must be in 1NF.
 - o All non-key attributes are fully functionally dependent on the primary key.
- Third Normal Form (3NF):
 - o Must be in 2NF.
 - o No transitive dependencies.
- Boyce-Codd Normal Form (BCNF):
 - o Must be in 3NF.
 - o Every determinant must be a candidate key.
- 25 Fourth Normal Form (4NF):
 - o Must be in BCNF.
 - o No multi-valued dependencies.
- Fifth Normal Form (5NF):
 - o Must be in 4NF.
 - o No join dependencies.

14 MCQ Table: subject_id (FK) Question option_1 option_2 option_3 option_4 correct
11 What is 20 the main purpose of the "System.out.println()" statement in Java? a)
Reading user input b) Declaring variables c) Printing output to the console d) Performing arithmetic operations option_3

Notification Table: notification_id Title message course_id (FK) date_time Fri Mar 29 12:13:18 GMT+05:30 2024 JAVA Attention

VIDEO Fri Mar 29 12:14:16 GMT+05:30 2024 Attention Students! Exciting news! A new video has just been uploaded for you all. Head over to your accounts to watch and learn. Don't miss out! MCQ JAVA QUESTION OPTION_1 OPTION_2 OPTION_3 OPTION_4 CORRECT What is 20 the main purpose of the "System.out.println()" statement in Java? a) Reading user input b) Declaring variables c) 12 Printing output to the console d) Performing arithmetic operations c) Printing output to the console ...

27 C/C++ QUESTION OPTION_1 OPTION_2 OPTION_3 OPTION_4 CORRECT What is the purpose of the "printf()" function in C? a) Reading input from the user b) Declaring variables c) Printing output to the console d) Performing arithmetic operations c) Printing output to the console ... PYTHON QUESTION OPTION_1 OPTION_2 OPTION_3

OPTION_3 OPTION_4 CORRECT What is Android Studio? a) An Android device b) An Android programming language c) An 17 Integrated Development Environment (IDE) for Android d) A type of Android application c) An Integrated Development Environment (IDE) for Android ... WEB QUESTION OPTION_1 OPTION_2 OPTION_3 OPTION_4

CORRECT What does the term "WWW" stand for in the context of web development? a) World Web Wide b) Web World Wide c) World Wide Web d) Wide Web World c) World

31 In this Gantt chart: • Each row represents a venture in the task. • Weeks are represented vertically. • '#' symbols imply the period of every venture. • Tasks that may run simultaneously are scheduled in parallel. • The vital route, that is the longest course through the project and 27 determines the shortest possible length for the venture, is highlighted by using the longest collection of connected responsibilities with none slack or flow. 12 In this example, the important path is: Requirement Analysis -> Planning -> Design -> Implementation/Coding -> Testing -> Deployment -> Maintenance.

address of the user name String Name of the user phone String Phone number of the user password String Password of the user id Integer Unique identifier for the user ... Course

10 Field Name Data Type Description courseCode String Code representing the course courseTitle String Title or name of the course course_icon String File path or URL to the icon representing course ... Subject Field Name Data Type Description subjectId String Identifier for the subject subjectName String Name of the subject courseId String Identifier for the course to which the subject belongs subject_icon String File path or URL to the icon representing the subject ... Action Field Name Data Type Description actionName String Name or description of the action actionPerformed Integer Unique identifier for the action ...

34 PDF Field Name Data Type Description subject_id Integer Identifier for the subject related to the PDF pdfs_link String Link or path to the PDF file ... Video Field Name Data Type Description subject_id String Identifier for the subject related to the video videos_link String Link or path to the video ... MCQ Field Name Data Type Description subject_id Integer Identifier for the subject of the MCQ question String The actual question presented to the user option_1 String First option for the multiple-choice question option_2 String Second option for the multiple-choice question option_3 String Third option for the multiple-choice question option_4 String Fourth option for the multiple-choice question correct String The correct option among the four options ... Notification 10 Field Name Data Type Description notification_id String Identifier for the notification title String Title of the notification message String Body or content of the notification course_id String Identifier for the associated course date_time String Date and time when the notification was created

a crucial position in validating the functionality and value of SS GYAN SYNC HUB. A dedicated team of testers systematically evaluates each characteristic and module of the platform, following predefined 6 take a look at instances and scenarios. Manual trying out encompasses various aspects, including user authentication, navigation, content material get entry to, shape submissions, and notification handling. Testers confirm that the

or attempting quizzes. o Progress tracking and feedback mechanisms ensure that users can monitor their learning journey and review their performance.

- Notification Management:
 - o The Notification module plays **21** a crucial role in keeping users informed about important updates and announcements within the platform.
 - o Users receive real-time notifications regarding new resources, upcoming quizzes, deadlines, and other relevant information, ensuring timely communication.

each teachers and students, permitting seamless account introduction and access to instructional assets. The Sign-In and Registration Form modules make sure stable consumer authentication, **6** at the same time as the Course and Subject modules enable easy navigation and choice of relevant academic packages and subjects.

- Effective Content Management: The Teacher module empowers educators to manipulate educational sources efficiently via the PDF Link, Video Link, and Quiz submodules.

180 net connectivity might also face challenges in having access to educational resources and participating in interactive activities inclusive of quizzes.

- Scalability Concerns: While the present day system structure supports **27** a significant number of users, scalability issues may get up as the consumer base grows. Adequate measures want to be carried out to make sure the device can accommodate growing user needs without compromising

Beginners. (n.d.). Retrieved from <https://www.youtube.com/watch?v=BGCBb8HPrOo>

- This tutorial series on YouTube provides step-by-step guidance for beginners in Android Studio, which can be beneficial for learning essential concepts and techniques.
- o Firebase Tutorial **17** for Android App Development. (n.d.). Retrieved from <https://www.youtube.com/watch?v=9kRgVxULbag>
- This tutorial covers Firebase integration in Android applications, offering practical demonstrations and explanations that can aid in implementing the backend functionalities of your system.

authenticate using their email and password. - New customers 12 will be able to sign in thru a Registration Form, offering important details inclusive of name, e-mail, telephone, and password. 2.2 Course Management - The device shall categorize courses such as Subjects shall be organized below respective guides. - Each subject shall have a unique identifier, situation call, associated direction ID, and an icon for visible illustration. 2.4. Resource Management (Teachers) - Teachers shall be capable of upload, delete, and edit PDF resources for subjects they're assigned to. - Teachers 12 will be able to control video assets associated with unique topics. - Teachers shall have the capacity to create and manipulate quizzes for assessing college students.

Education Management System goals to offer a complete platform for academic content material transport and assessment. By adhering to the outlined useful and nonpractical necessities, the system will make certain a seamless person enjoy 6 at the same time as preserving safety, reliability, and scalability.

be redirected lower back to the login page. Now, proceed with the login technique. 4. Main Interface: Upon successful login, you may be directed to the primary interface. 5. Actions for Teachers: a. If you're 14 logged in as a teacher: i. You'll see alternatives to feature notifications, PDF links, video playlist hyperlinks, and quizzes.

video playlist hyperlinks for supplementary learning materials. v. Add/edit quizzes to evaluate your students' information. 6. Actions for Students: a. 14 If you are logged in as a pupil: i. You'll see notifications furnished via your teachers. ii. Access PDF hyperlinks for extra study materials. iii. Watch video playlists shared by using your instructors for stronger the login page. In Summary: Our interactive studying platform offers a seamless revel in for both teachers and students. Teachers can without difficulty control path substances and checks, 6 at the same time as college students have get admission to to a variety of assets to enhance their studying adventure. With consumer-pleasant interfaces and sturdy features, our platform objectives to facilitate a dynamic and attractive instructional enjoy for