Chapter 1

INTRODUCTION

1.1 PROBLEM DEFINITION:

Education system forms the backbone of every nation. And hence it is important to provide a strong educational foundation to the young generation to ensure the development of open-minded global citizens securing the future for everyone. Advanced technology available today can play a crucial role in streamlining education-related processes to promote solidarity among students, teachers, parents and the school staff. Education is central to development. It is one of the most powerful instruments for reducing poverty and inequality and lays a foundation for sustained economic growth. With this aim currently our government has given special emphasis to the educational sector and school improvement activities such as continuous professional development for teachers, training and upgrading teachers and capacitating schools with manpower and materials are among the major actions which have been taken in both primary and secondary schools. In order to facilitate and simplify these actions one of the major tools is to have automated school management system.

1.2 PURPOSE:

1. To make learning easy.
2. To design a highly-reactive web application.
3. To make software fast in processing, with a good user interface.
4. To allow users to share their experiences with others.
5. For easy record of subjects.

1.3 SCOPE:

This scope of school management is very vast. It includes everything regarding the efficient functioning of the educational institution, securing the greatest benefit to the greatest number through an adoption of practical measures. It interprets and clarifies the functions and the activities of an educational program in fruitful relationships and harmonizes their mutual action. It ensures sound planning, good direction and efficient and systematic execution.

Chapter 2

SYSTEM ANALYSIS

2.1 Existing system explanation:

- The existing system that is available at https://www.teachmint.com, is a school management system that does a lot of basic crud operations.
- The biggest problem with these software solutions is that these were written in pre-pandemic era and hence minimal to zero support for virtual or hybrid education.
- These are mostly written with school management and teacher in mind and not the students and hence in these virtual/hybrid education days these fail terribly.
- The UI is very complex and sometimes to tiring to use, not only that but not all features are available free-to-use.
- Almost none of these softwares have support for live classes or take-home assignments.
- These are generally slow and prone to bugs which are difficult to fix because of old technologies.

2.2 Proposed system explanation:

- The system in question solves multiple of the above mentioned problem, because it is designed and made with hybrid education in mind.
- We have put in extra care to make sure that the UI in model and intuitive to increase the attention span of students on the system.
- Because the current system is made with hybrid education in mind we have a lot of features focused towards remote education.
- Our platform supports live online classes from within the system at top quality.
- We have support for notes and classes archive where the notes made by professors can be uploaded and accessed by students to further understand the topics.
- These notes and classes have a comments section for students and teachers to discuss the topic and clear doubts.
- Keeping it simple is the basic aim of Skool it.
- Our quiz module is based on the keep it simple vision, where the teacher uploads a question paper as pdf and then students can upload their solutions for him/her to check and then finally a solution key can be uploaded

2.3 System requirements explanation:

2.3.1 Hardware:

- AMD Ryzen 5 4500U
- Fast and high bandwidth internet connection.
- 8 GB RAM SSD
- AMD Radeon Graphics
- Operating system like Windows/Linux/MacOS.
- Computer/PC with good storage and good ram.

2.3.2 Software:

- Frontend= HTML, CSS, JavaScript, Vue.js, Bulma
- Backend= Firebase OAuth, Firestore, Firebase Storage, Cloud Functions
- Hoisting= Firebase
- Version Control System= Git/GitHub
- Database= No SQL.

Chapter 3

SYSTEM DESIGN

The system design involves the production of technical and visual prototypes. This stage has some non-technical aspects such as gathering of web contents. For the server-side programming and other technical aspects of the design emphasis will be laid on such design concepts and principles as effective modularity, information hiding and stepwise elaboration. The goal is to make the system easier to adapt, enhance, test, and use.

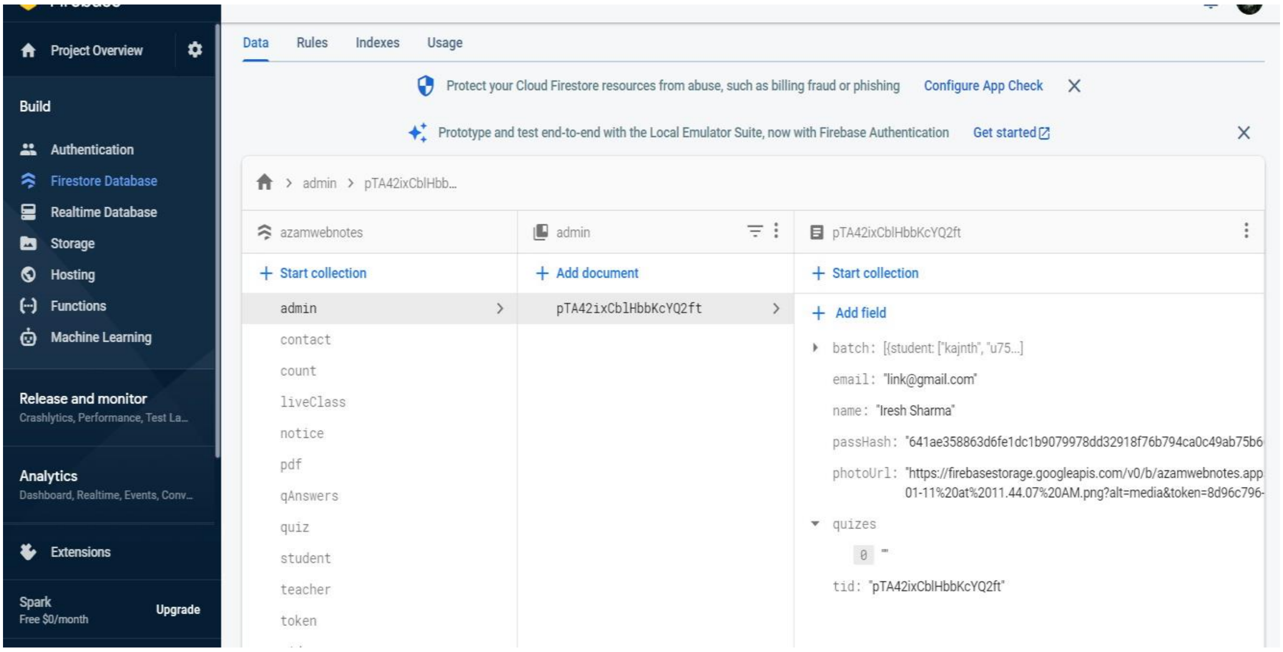
3.1 Input Design:

The user must sign up if he is new to the system by using their personal mail-id. Once the user has signed up, they can login to their account and explore the top-bar of the homepage. They will also be able to see the all the publicly available posts in the Homepage. The user can know when a particular post was created and by whom, to get the latest data which is available in the vast ocean that is the internet.

The user will be able to edit their own posts and comments only, delete too. They won't be able to edit or delete the posts of other users. The inputs they will be providing is the name, email-id and a password to login and later can join any ongoing live class or can also access the lectures and notes from before.

3.2 Database Design:

Firebase is a Backend-as-a-Service (Baas). It provides developers with a variety of tools and services to help them develop quality apps, grow their user base, and earn profit. It is built on Google's infrastructure. Firebase is categorized as a NoSQL database program, which stores data in JSON-like documents. Firebase can power your app's backend, including data storage, user authentication, static hosting, and more. Focus on creating extraordinary user experiences. You can also connect Firebase to your existing backend using the server-side libraries or our REST API.



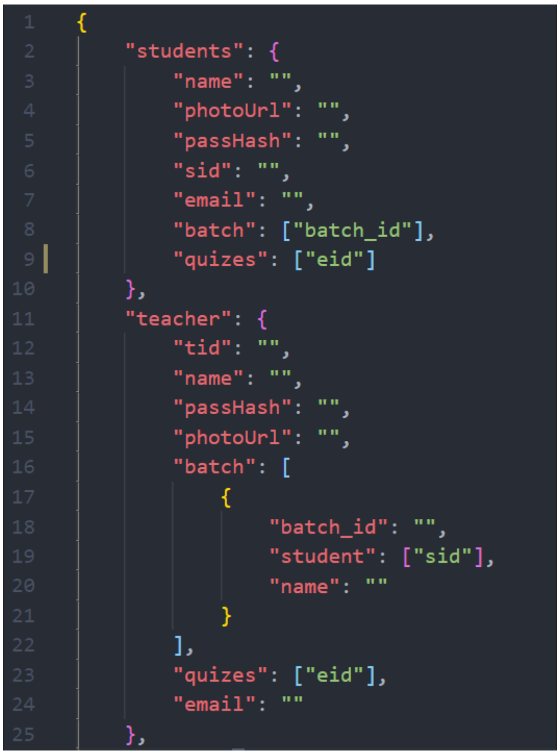
```
batch: [{student: ["kajnth", "u75...]
email: "link@gmail.com"
name: "Iresh Sharma"
passHash: "641ae358863d6fe1dc1b9079978dd32918f76b794ca0c49ab75b6
photoUrl: "https://firebasestorage.googleapis.com/v0/b/azamwebnotes.app
          01-11%20at%2011.44.07%20AM.png?alt=media&token=8d96c796-
quizes
    0 ""
tid: "pTA42ixCblHbbKcYQ2ft"
```

Fig. 3.1(Student Management System Database)

❖ Database contains several collections like admin, contact, student, token etc.

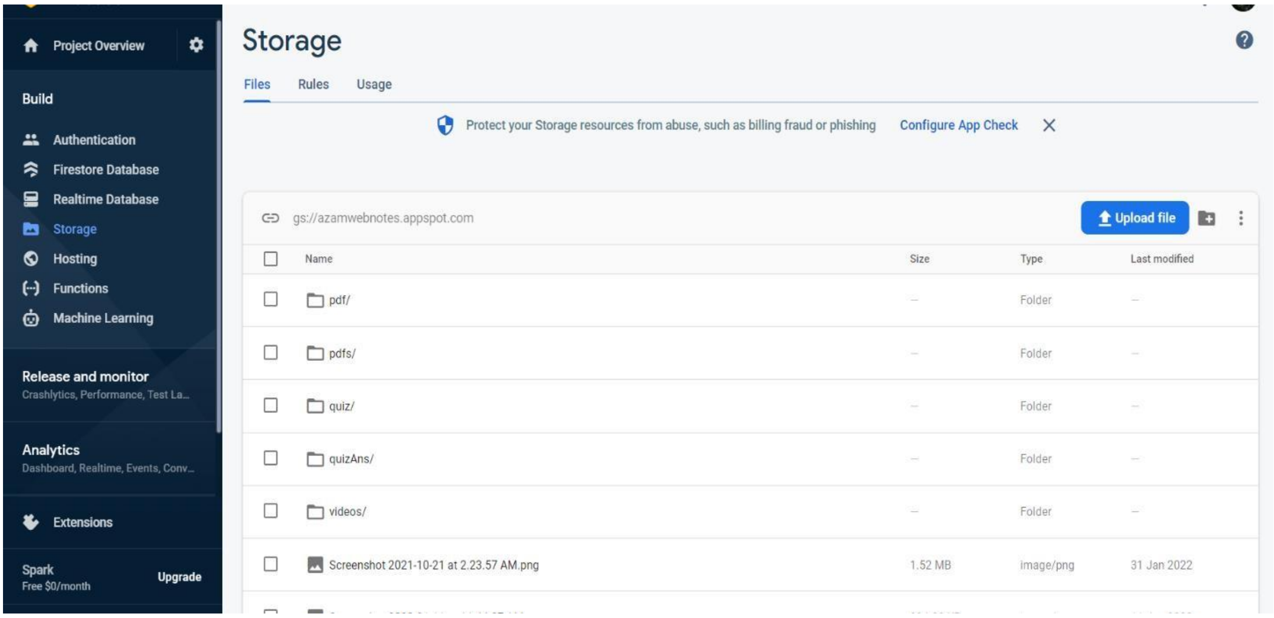
Collection Schema:

This collection is used to store data of Students and Teachers.



```
{
    "students": {
        "name": "",
        "photoUrl": "",
        "passHash": "",
        "sid": "",
        "email": "",
        "batch": ["batch_id"],
        "quizes": ["eid"]
    },
    "teacher": {
        "tid": "",
        "name": "",
        "passHash": "",
        "photoUrl": "",
        "batch": [
            {
                "batch_id": "",
                "student": ["sid"],
                "name": ""
            }
        ],
        "quizes": ["eid"],
        "email": ""
    },
```

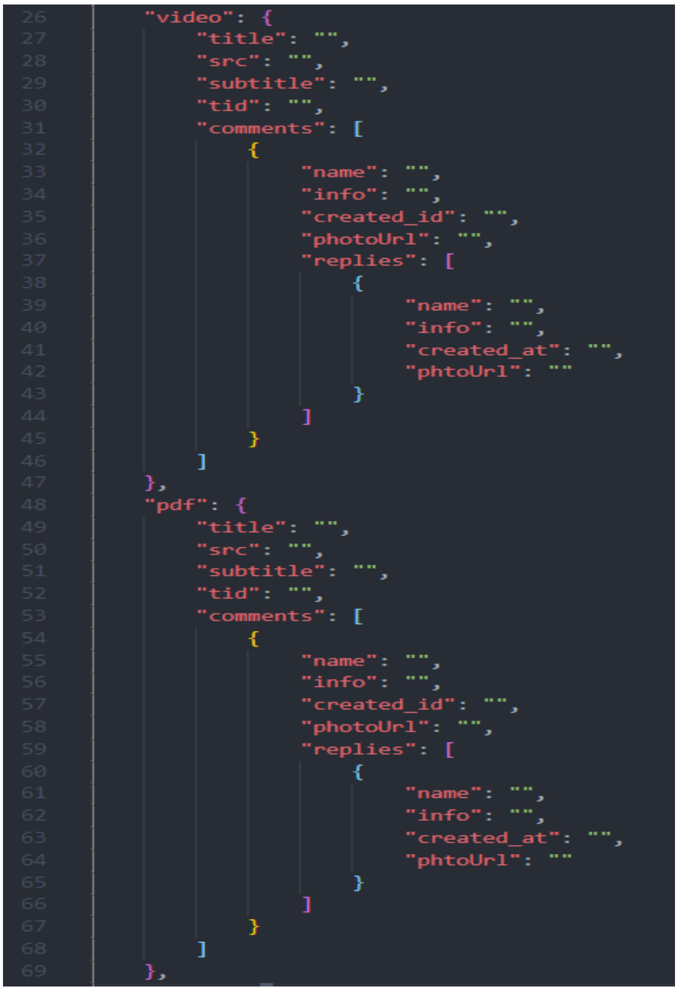
Fig. 3.2(Schema1)



gs://azamwebnotes.appspot.com

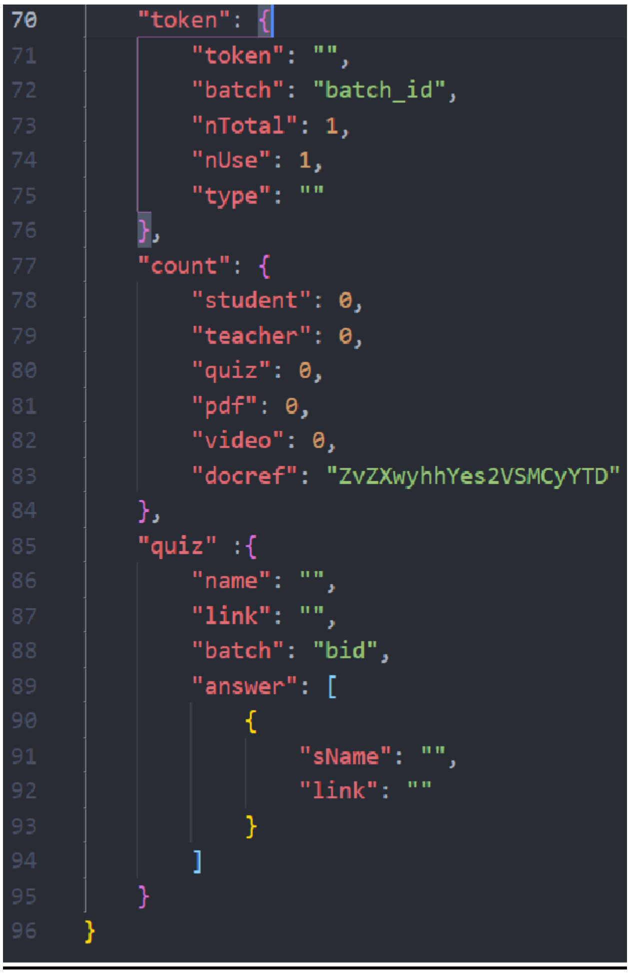
□ Name	Size	Type	Last modified
□ pdf/	—	Folder	—
□ pdfs/	—	Folder	—
□ quiz/	—	Folder	—
□ quizAns/	—	Folder	—
□ videos/	—	Folder	—
□ Screenshot 2021-10-21 at 2.23.57 AM.png	1.52 MB	image/png	31 Jan 2022

Fig. 3.3(These are all the files present in the Firestore overview. All the data present in the website is being stored here.)



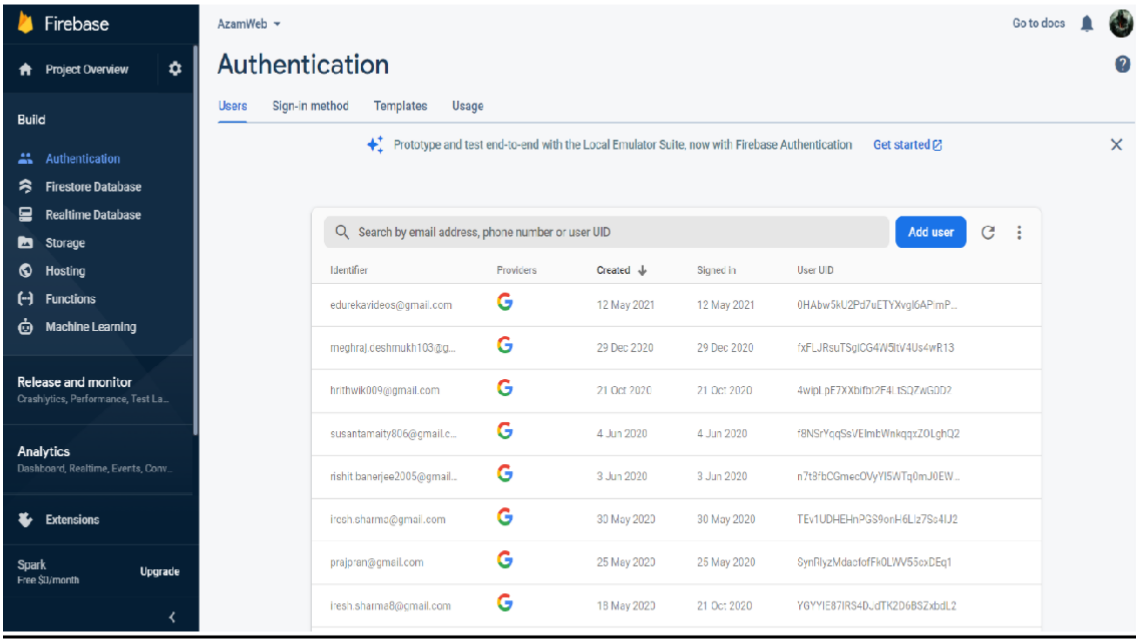
```
    "video": {
        "title": "",
        "src": "",
        "subtitle": "",
        "tid": "",
        "comments": [
            {
                "name": "",
                "info": "",
                "created_id": "",
                "photoUrl": "",
                "replies": [
                    {
                        "name": "",
                        "info": "",
                        "created_at": "",
                        "phtoUrl": ""
                    }
                ]
            }
        ]
    },
    "pdf": {
        "title": "",
        "src": "",
        "subtitle": "",
        "tid": "",
        "comments": [
            {
                "name": "",
                "info": "",
                "created_id": "",
                "photoUrl": "",
                "replies": [
                    {
                        "name": "",
                        "info": "",
                        "created_at": "",
                        "phtoUrl": ""
                    }
                ]
            }
        ]
    },
```

Fig. 3.4(Schema2)



```
    "token": {
        "token": "",
        "batch": "batch_id",
        "nTotal": 1,
        "nUse": 1,
        "type": ""
    },
    "count": {
        "student": 0,
        "teacher": 0,
        "quiz": 0,
        "pdf": 0,
        "video": 0,
        "docref": "ZvZXwyhhYes2VSMCyYTD"
    },
    "quiz" :{
        "name": "",
        "link": "",
        "batch": "bid",
        "answer": [
            {
                "sName": "",
                "link": ""
            }
        ]
    }
}
```

Fig. 3.5(Schema 3)

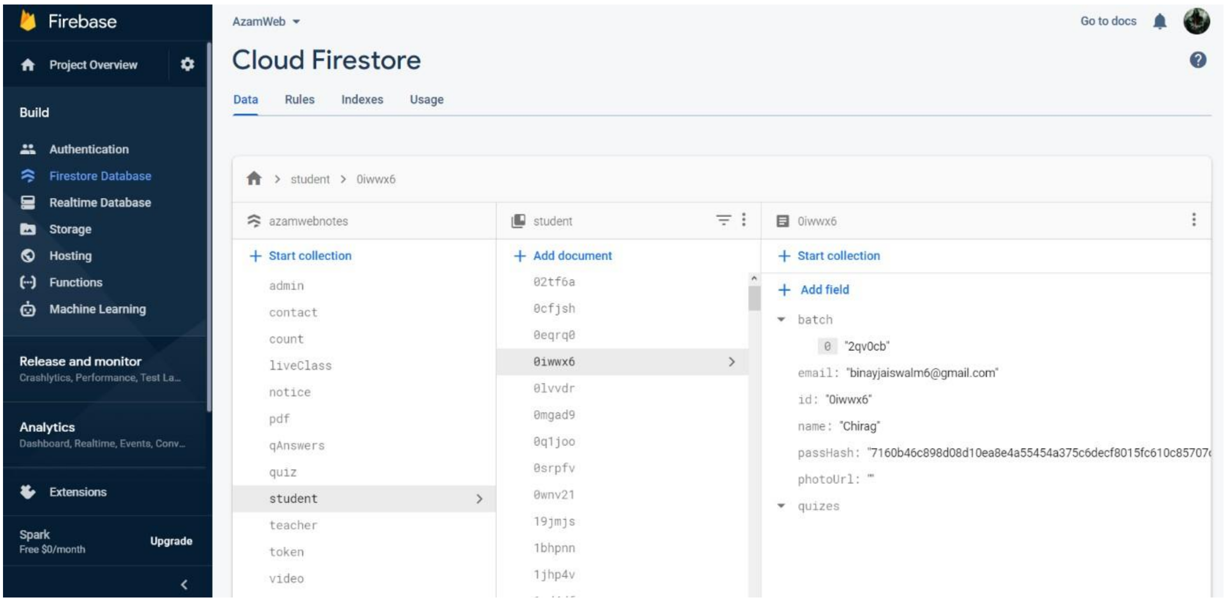


Authentication

Prototype and test end-to-end with the Local Emulator Suite, now with Firebase Authentication

Identifier	Providers	Created	Signed in	User UID
edurekavideos@gmail.com	G	12 May 2021	12 May 2021	0HAbw5kU2Pd7uETYXvgl6APimP...
meghraj.ceshmukh103@g...	G	29 Dec 2020	29 Dec 2020	fxFLJRsuTSgICG4W5ltV4Us4wR13
hrithwik009@gmail.com	G	21 Oct 2020	21 Oct 2020	4wipLpF7XXbifbt2F4LtSQ7wGDD2
susantamaity806@gmail.c...	G	4 Jun 2020	4 Jun 2020	f8NSrYqqSsVElmbWnkqqxZ0LghQ2
rishit.banerjee2005@gmail...	G	3 Jun 2020	3 Jun 2020	n7t3fbCGmecOVyYI5WTq0mJ0EW...
iresh.sharma@gmail.com	G	30 May 2020	30 May 2020	TEv1UDHEHnPGS9onH6LIz7Ss4IJ2
prajpran@gmail.com	G	25 May 2020	25 May 2020	SynRlyzMdacfofFk0LWV55cxDEq1
iresh.sharma8@gmail.com	G	18 May 2020	21 Oct 2020	YGYYIE87IRS4DJdTK2D6BSZxbdL2

Fig. 3.6(Here we can see who all have accessed the database and when)



azamwebnotes: admin, contact, count, liveClass, notice, pdf, qAnswers, quiz, student, teacher, token, video

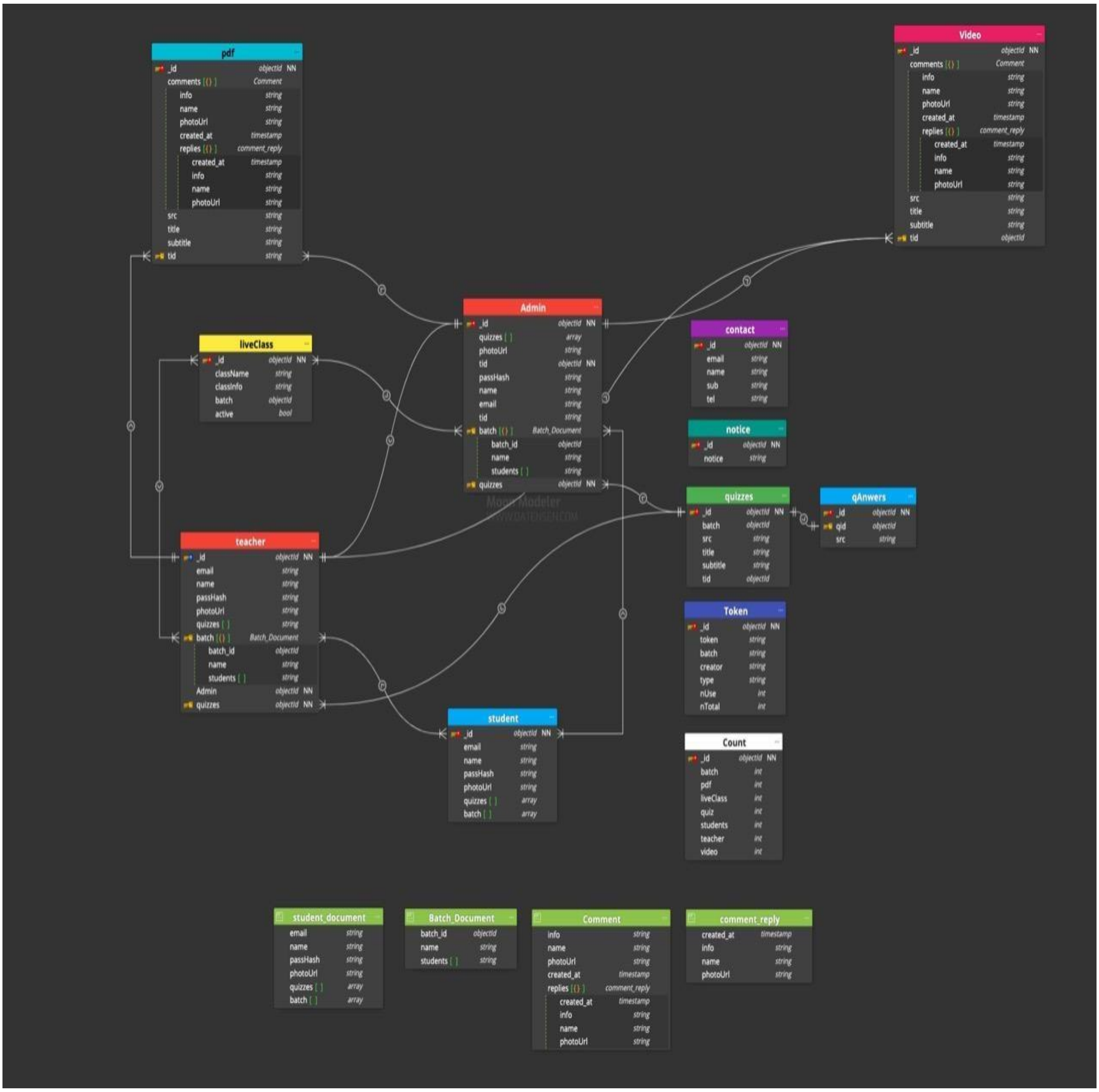
student: 02tf6a, 0cfjsh, 0eqrq0, 0iwwx6, 0lvvdr, 0mgad9, 0q1joo, 0srpfv, 0wnv21, 19jmjs, 1bhpnn, 1jhp4v

```
0iwwx6
batch
    0 "2qv0cb"
email: "binayjaiswalm6@gmail.com"
id: "0iwwx6"
name: "Chirag"
passHash: "7160b46c898d08d10ea8e4a55454a375c6decf8015fc610c85707
photoUrl: ""
quizes
```

Fig. 3.7(we can see the student collection with all the documents present in it)

- Firebase is best useful here as it is best for dynamic use.
- The need to create multiple collections is avoided as each collection can be accessed as per the need of the developer.

3.3 E-R Diagram:



pdf		
_id	objectid	NN
comments [{}]	Comment	
info	string	
name	string	
photoUrl	string	
created_at	timestamp	
replies [{}]	comment_reply	
created_at	timestamp	
info	string	
name	string	
photoUrl	string	
src	string	
title	string	
subtitle	string	
tid	string	

Video		
_id	objectid	NN
comments [{}]	Comment	
info	string	
name	string	
photoUrl	string	
created_at	timestamp	
replies [{}]	comment_reply	
created_at	timestamp	
info	string	
name	string	
photoUrl	string	
src	string	
title	string	
subtitle	string	
tid	objectid	

Admin		
_id	objectid	NN
quizzes [ ]	array	
photoUrl	string	
tid	objectid	NN
passHash	string	
name	string	
email	string	
tid	string	
batch [{}]	Batch_Document	
batch_id	objectid	
name	string	
students [ ]	string	
quizzes	objectid	NN

liveClass		
_id	objectid	NN
className	string	
classInfo	string	
batch	objectid	
active	bool	

contact		
_id	objectid	NN
email	string	
name	string	
sub	string	
tel	string	

notice		
_id	objectid	NN
notice	string	

quizzes		
_id	objectid	NN
batch	objectid	
src	string	
title	string	
subtitle	string	
tid	objectid	

qAnwers		
_id	objectid	NN
qid	objectid	
src	string	

teacher		
_id	objectid	NN
email	string	
name	string	
passHash	string	
photoUrl	string	
quizzes [ ]	string	
batch [{}]	Batch_Document	
batch_id	objectid	
name	string	
students [ ]	string	
Admin	objectid	NN
quizzes	objectid	NN

Token		
_id	objectid	NN
token	string	
batch	string	
creator	string	
type	string	
nUse	int	
nTotal	int	

student		
_id	objectid	NN
email	string	
name	string	
passHash	string	
photoUrl	string	
quizzes [ ]	array	
batch [ ]	array	

Count		
_id	objectid	NN
batch	int	
pdf	int	
liveClass	int	
quiz	int	
students	int	
teacher	int	
video	int	

student_document	
email	string
name	string
passHash	string
photoUrl	string
quizzes [ ]	array
batch [ ]	array

Batch_Document	
batch_id	objectid
name	string
students [ ]	string

Comment	
info	string
name	string
photoUrl	string
created_at	timestamp
replies [{}]	comment_reply
created_at	timestamp
info	string
name	string
photoUrl	string

comment_reply	
created_at	timestamp
info	string
name	string
photoUrl	string

The above E-R diagram represents the E-R diagram for Student Management System.

It consists of 12 entities namely admin, teacher, student, pdf, video, live classes, token, count, quizzes, notice and Question & Answers.

The various relationships are mentioned below:

1. **Teacher to student**: Every teacher has a list of batches, each having a list of students stored by using foreign keys.

2. **Admin to teacher**: Since the admin is also a teacher hence the admin document has a key tid to reference the teacher’s collection.

3. **Admin to student**: Since admin is also a teacher he/she too has a batches array each having a list of students storing ids to references the students collection.

4. **Teacher to quizzes**: Each teacher has a list of quizzes to store the qid of the document in the quizzes collection.

5. **Admin to quizzes**: Since admin is also a teacher, it too has a list of quizzes to store the qid of the document in the quizzes collection.

6. **Live Classes to teacher**: Each live class is associated with a batch and hence live class document contains a batch_id key referencing to the teacher collection.

7. **pdf to teacher**: Since a pdf is uploaded by a teacher hence the pdf has a tid key referencing a teacher.

8. **Video to teacher**: Since a pdf is uploaded by a teacher hence the pdf has a tid key referencing a teacher.

9. **qAnswer to quiz**: The qAnswer documents have a qid each to references the quiz to which the answer is saved.

Chapter 4

SYSTEM IMPLEMENTATION

4.1 Implementation:

As discussed above, the backend of the website is made using Firebase OAuth, Firestore, Firebase Storage, Cloud Functions and the front is made using HTML, CSS, JavaScript, Vue.js, Bulma. The details of these libraries are given below. Implementation environment is in Visual Studio which provides the user-friendly Integrated Development Environment.

The functions that this system will be implementing are:

- Create an account for a new user.
- Store the data such as mail-id and password.
- If the mail-id/username is same as other users, then a pop-up occurs saying that the mail-id/username already exists.
- Updates the data in the database every time an account is created.
- Once account is created the user can access all the resources provided by the admin.
- The user will be able to view others too.

4.2 Tools used for implementation:

Cascading Style Sheets (CSS):

Cascading Style Sheets, fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page.

CSS is easy to learn and understand, but it provides powerful control over the presentation of an HTML document.

CSS saves time: You can write CSS once and reuse the same sheet in multiple HTML pages.

Easy Maintenance: To make a global change simply change the style, and all elements in all the webpages will be updated automatically.

Search Engines: CSS is considered a clean coding technique, which means search engines won't have to struggle to "read" its content.

Superior styles to HTML: CSS have a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.

Offline Browsing: CSS can store web applications locally with the help of an offline cache. Using this we can view offline websites.

JavaScript:

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities. JavaScript is one of the three-core technologies of the world wide web content production. It is used to make web pages interactive and provide online programs, including video games. The majority of websites employ it, and all modern web programs support it without the need for plug-ins by means of built-in JavaScript engine.

Vue JS:

Vue JS is an open-source progressive JavaScript framework used to develop interactive web interfaces. It is one of the famous frameworks used to simplify web development. Vue JS focusses on the view layer. It can be easily integrated into big projects for front-end development without any issues. The installation for Vue JS is very easy to start with. Any developer can easily understand and build interactive web interfaces in a matter of time. Vue JS is created by Evan You, an ex-employee from Google.

Node JS:

Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux. Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

Bulma:

Bulma is a light weight, modern CSS framework, which depends on flexbox module (it is used for developing responsive layout structure and fancy designs). The overview of Bulma includes getting started with Bulma, responsiveness (you can view the web pages on different devices such as desktops, tablets, and phones), colors, functions to define colors and values, and mix ins (group of CSS properties that allow you to use properties of one class for another class) in the Bulma.

Firebase Authentication:

Most apps need to know the identity of a user. Knowing a user's identity allows an app to securely save user data in the cloud and provide the same personalized experience across all of the user's devices. Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more. Firebase Authentication integrates tightly with other Firebase services, and it leverages industry standards like OAuth 2.0 and OpenID Connect, so it can be easily integrated with your custom backend.

Firestore:

Firestore is a NoSQL document database built for automatic scaling, high performance, and ease of application development. While the Firestore interface has many of the same features as traditional databases, as a NoSQL database it differs from them in the way it describes relationships between data objects. Firebase Realtime Database, it keeps your data in sync across client apps through real-time listeners and offers offline support for mobile and web so you can build responsive apps that work regardless of network latency or Internet connectivity. Firestore also offers seamless integration with other Firebase and Google Cloud products, including Cloud Functions.

Firebase Storage:

The Firebase SDK lets developers to work with Firebase programmatically, including cloud storage functionality. It enables them to download and upload files from and to clients directly. It also includes features that enable transfers to be resumed or retried in cases of poor connectivity. When Google Cloud Storage is incorporated into Firebase apps, you gain access to Google security measures and the ability to secure any uploads or downloads in your app. Through the SDK you can also manage your media and access it directly from your storage account. Integration is supported for Android, C++, iOS, Unity, and Web apps. Although Firebase Storage uses a real-time database you can access data via a familiar file/folder system. When accessing files, you simply need to call the reference where your file is stored. References also enable you to control where files are stored and how files are labeled.

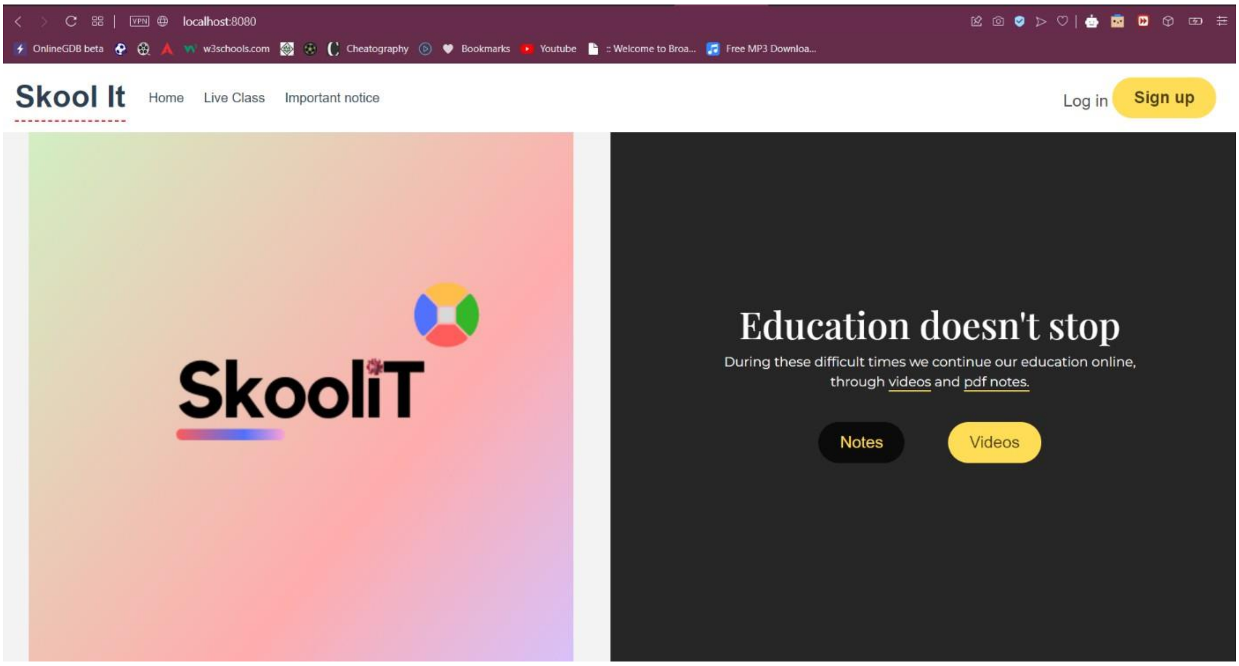
GitHub:

GitHub is a website and cloud-based service that helps developers store and manage their code, as well as track and control changes to their code. To understand exactly what GitHub is, you need to know two connected principles: Version control & Git. Version control helps developers track and manage changes to a software project's code. As a software project grows, version control becomes essential. Git is a distributed version control system**,** which means that the entire codebase and history is available on every developer's computer, which allows for easy branching and merging.

Chapter 5

SYSTEM TESTING

All the test cases successfully give the expected results and are as follows:

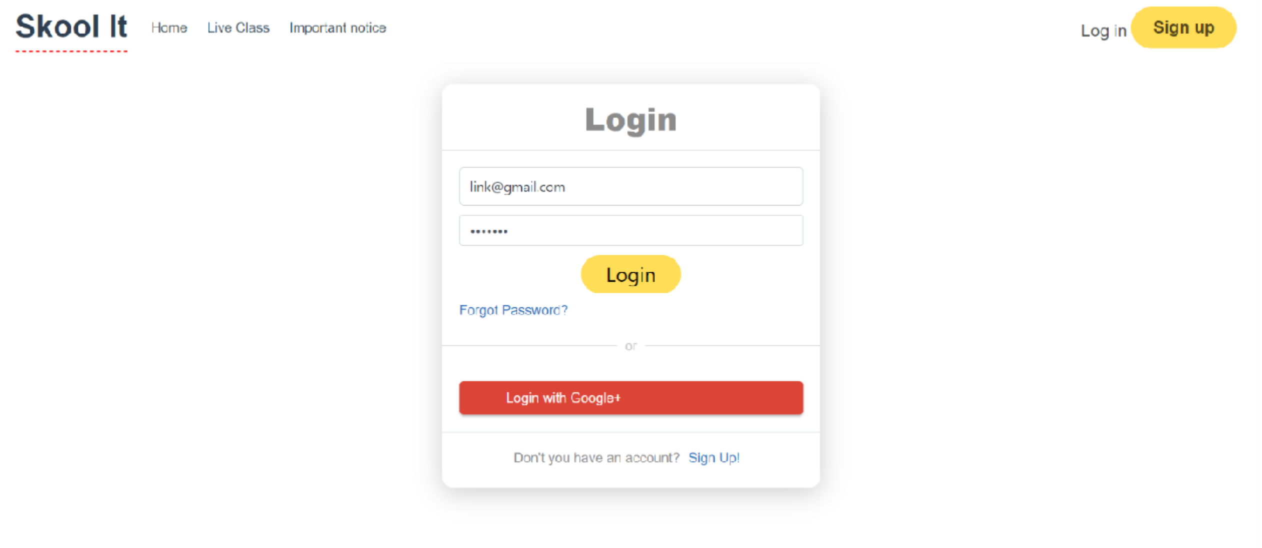


Education doesn't stop

During these difficult times we continue our education online, through videos and pdf notes.

Fig. 5.1(Home page 1)

- This is the page that shows when a person opens our site.
- Testing is done extensively for login page.
- A user needs to have an account to login.
- Testing is done using our demo email IDs.



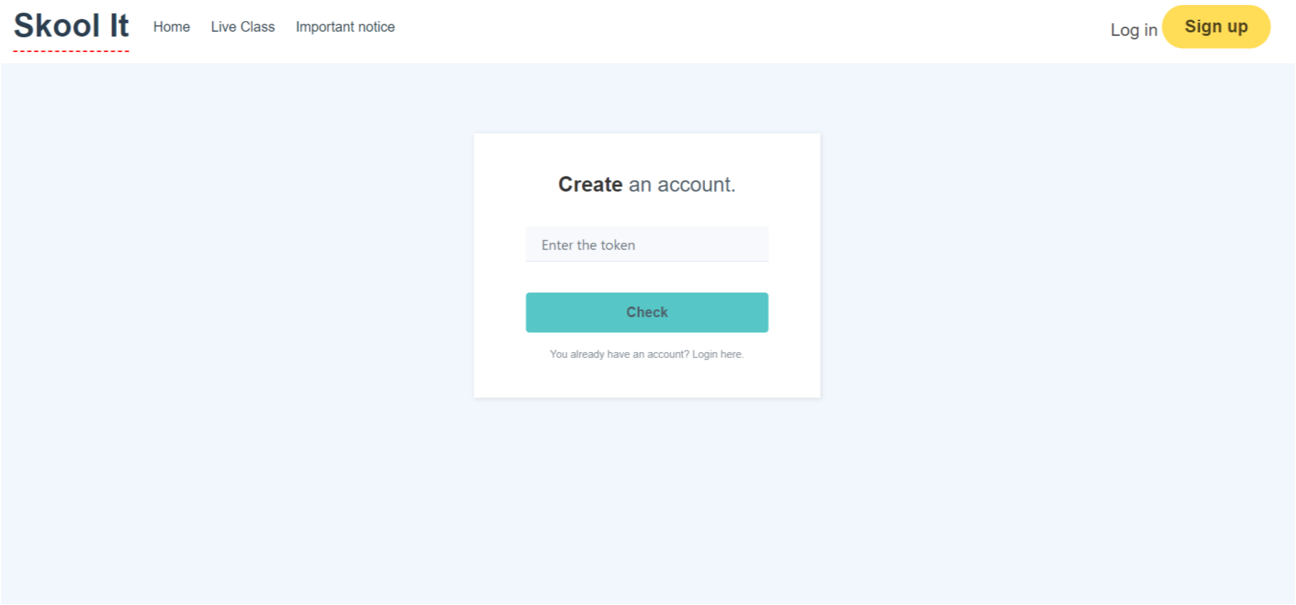
link@gmail.com

•••••••

Don't you have an account? Sign Up!

Fig. 5.2(Login page)

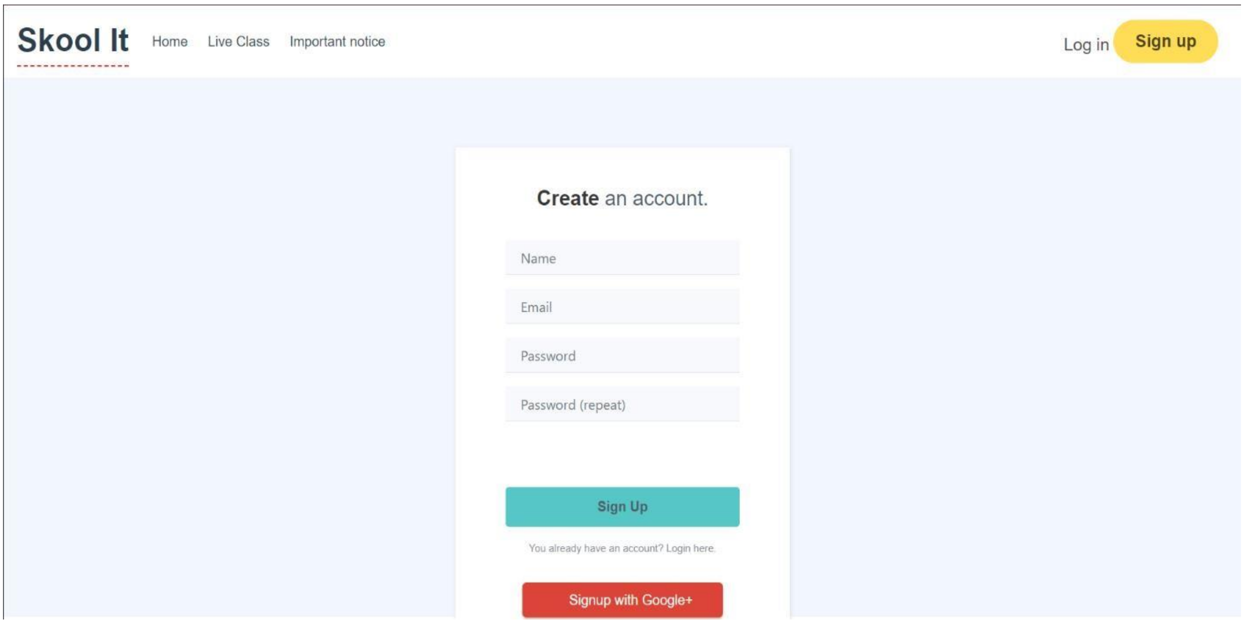
- We can enter the credentials to log into our respective accounts.



You already have an account? Login here.

Fig. 5.3(Signing up into our site)

- The sign up for a new account page. A person needs to have an account in order to access the site, they can create a new account here. Firstly, admin has to provide a unique token which stops outsiders from registering to this website. After the token is verified, user can go ahead with the process of creating his/her account.

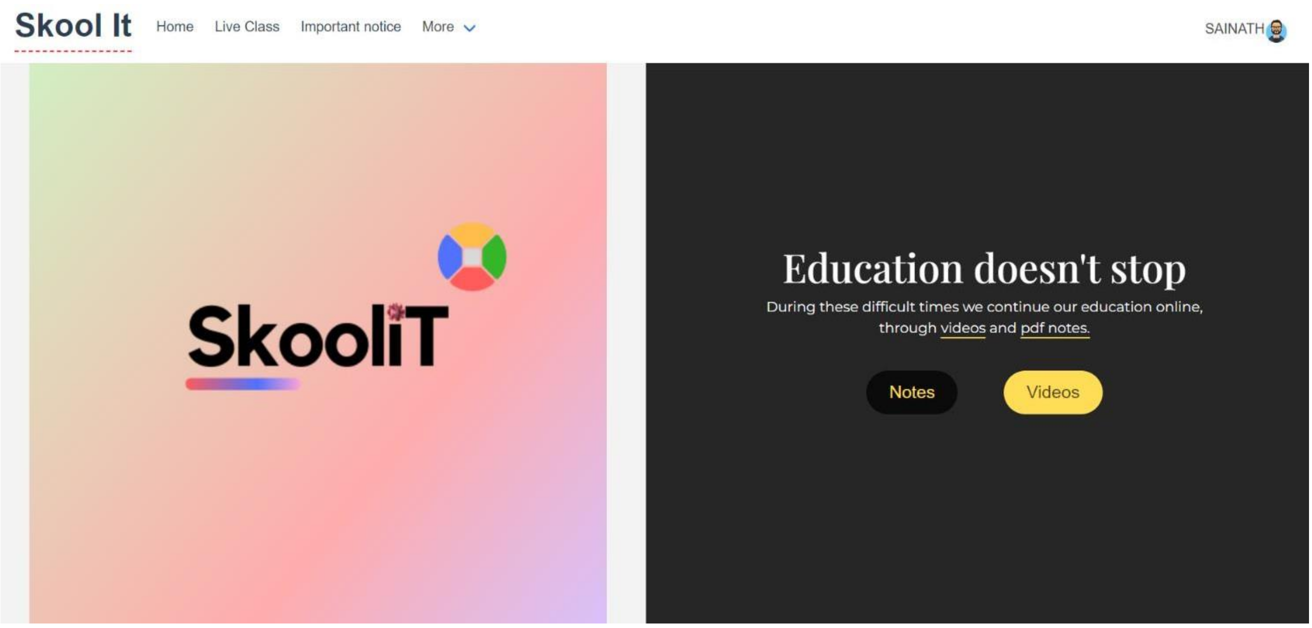


Create an account.

You already have an account? Login here.

Fig. 5.4(Creating an account after token is given by the admin to register)

- If a user tries to create an account with already existing mail-id or username, then a message occurs saying that "Username or email-ID already exists".

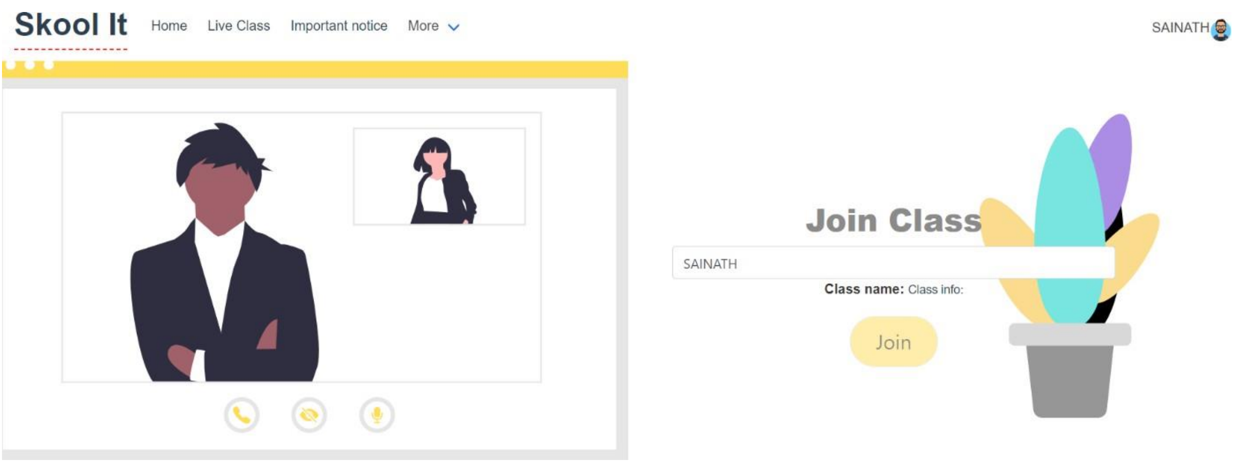


Education doesn't stop

During these difficult times we continue our education online, through videos and pdf notes.

Fig. 5.5(Home page 2)

- This page shows up when a user Logs in.
- It displays the nav-bar that contains the Home, Live Class, Important Notice and more (notes, videos and quiz).

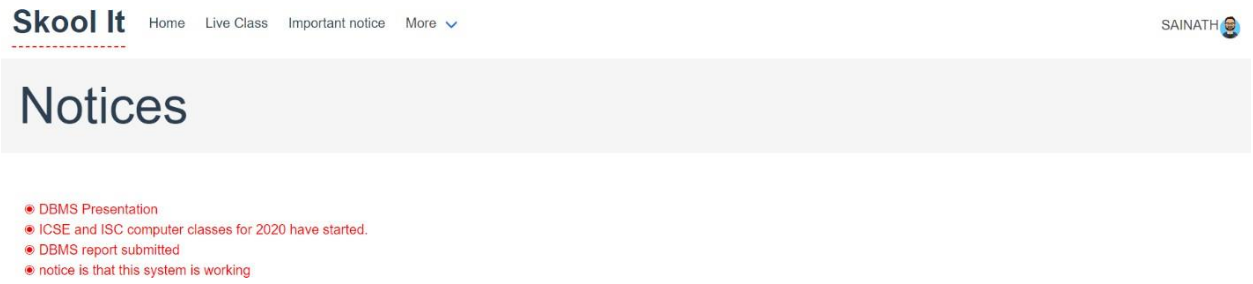


SAINATH

Class name: Class info:

Fig. 5.6(Live Class section)

- Displays if there are any ongoing live classes. Name cannot be changed while joining the class.
- Automatically detects any running classes for your batch.



Notices

- DBMS Presentation
- ICSE and ISC computer classes for 2020 have started.
- DBMS report submitted
- notice is that this system is working

Fig. 5.7(Important Notice Section)

- It’s an online Notice board for the users.

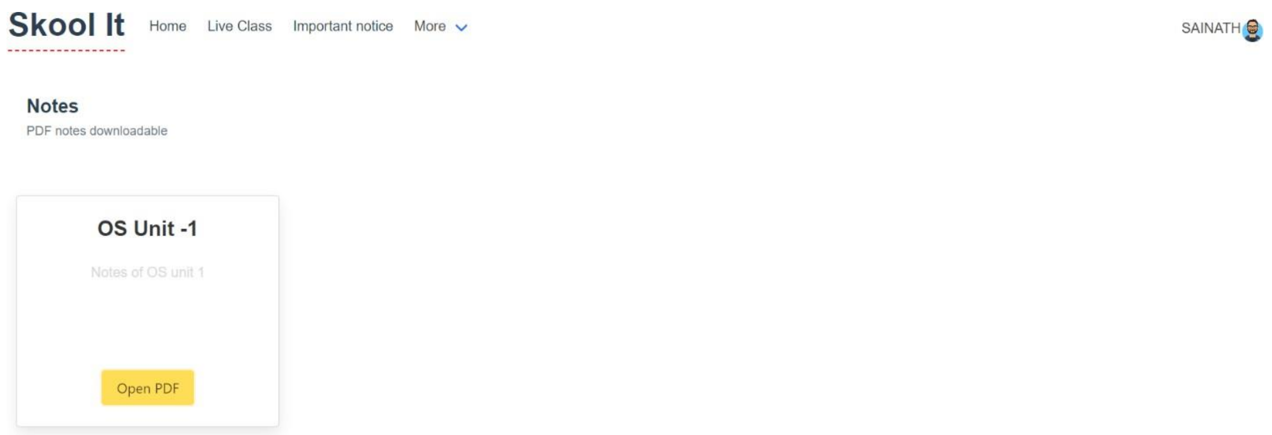


Fig. 5.8(Notes Section [Under more])

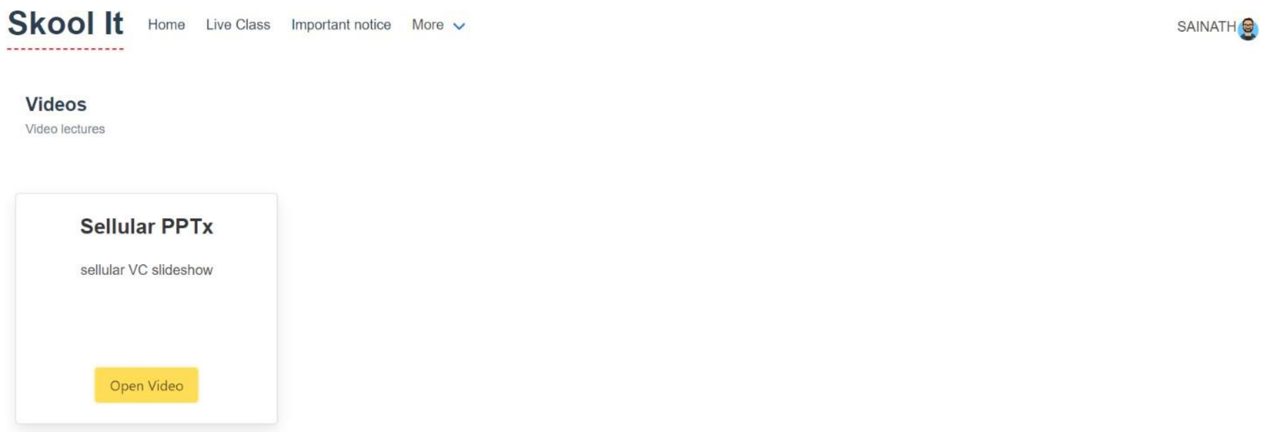
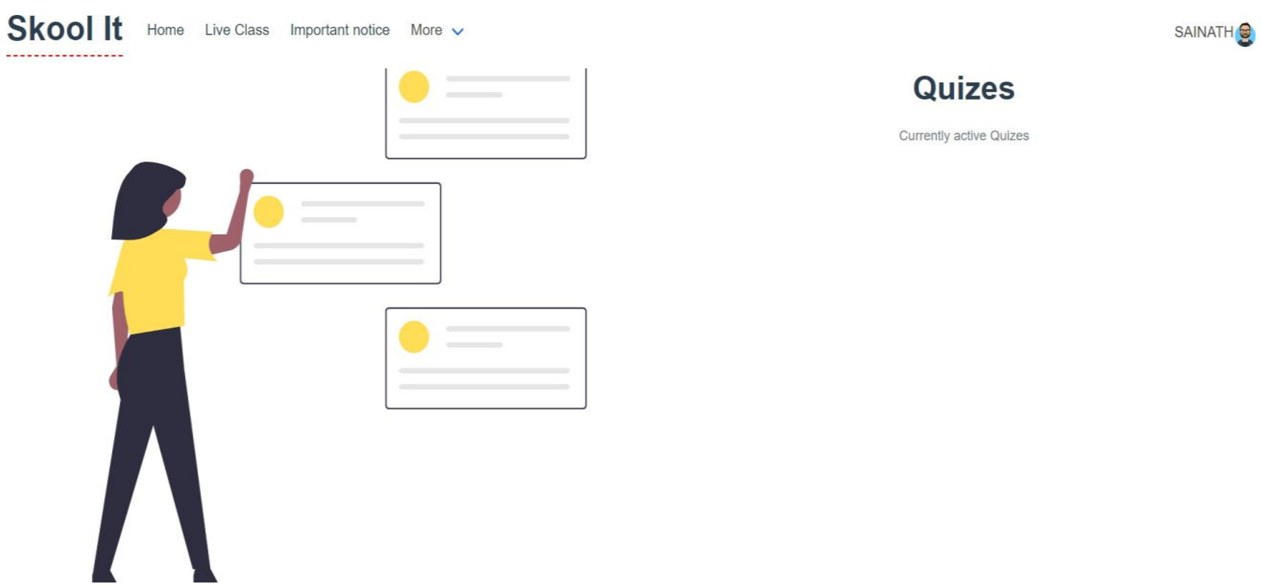


Fig. 5.9(Video Section [Under more])

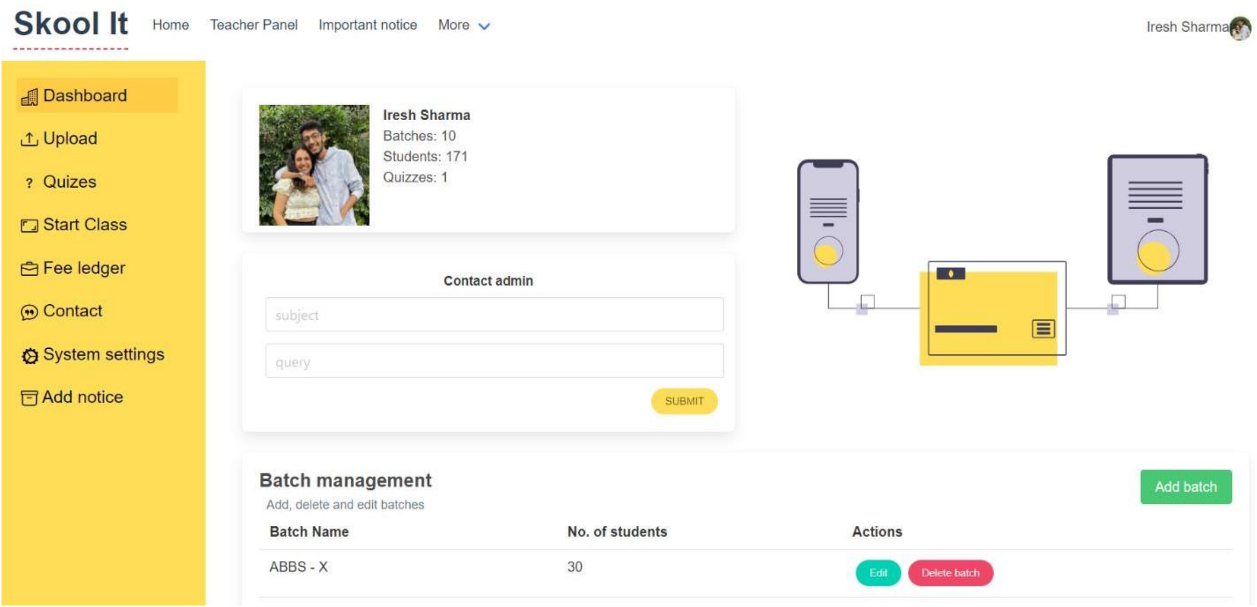
- Here, the users can check the videos uploaded by the admin like previous lectures etc. to make the learning process easier.



Quizes

Currently active Quizes

Fig. 5.10(Quiz Section [Under more])



Iresh Sharma
Batches: 10
Students: 171
Quizzes: 1

Contact admin

subject

query

Batch management

Add, delete and edit batches

Batch Name	No. of students	Actions
ABBS - X	30	Edit Delete batch

Fig. 5.11(Teacher Panel)

- This menu item is only available to the admin who handles the whole system.

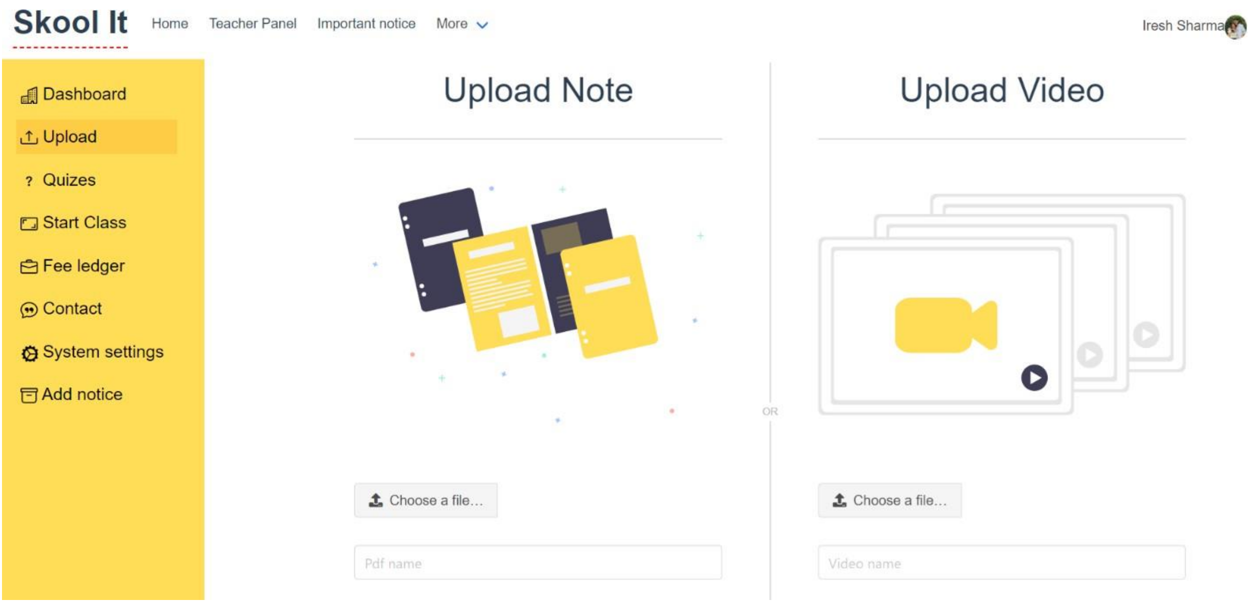
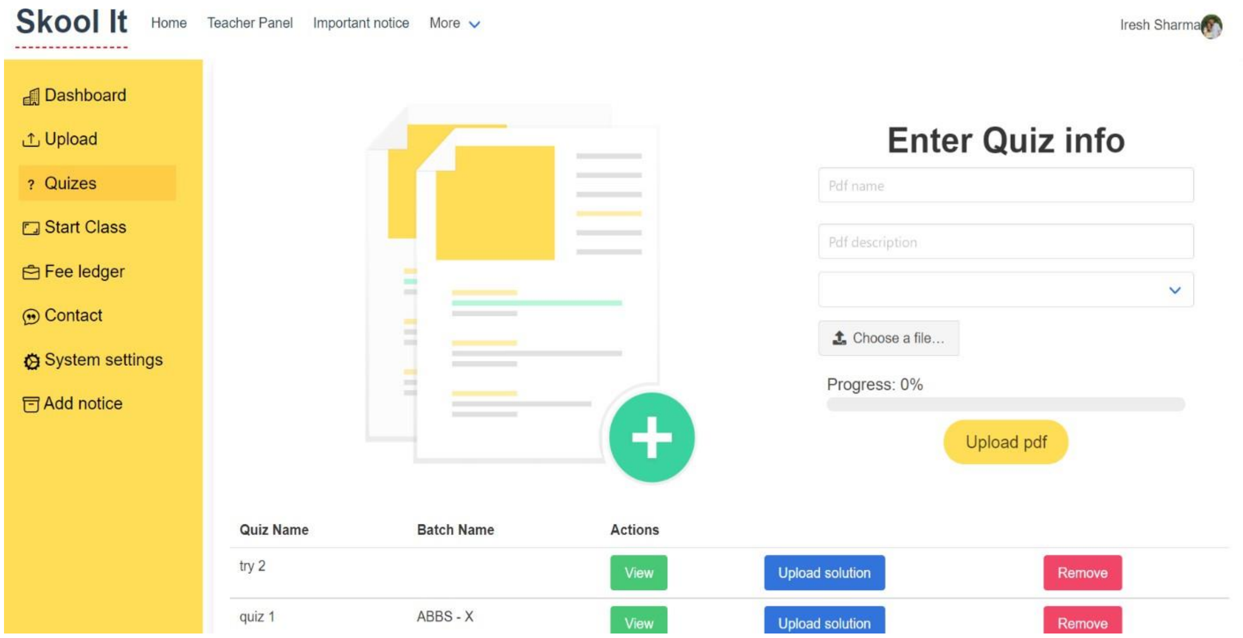


Fig. 5.12(Upload Option)

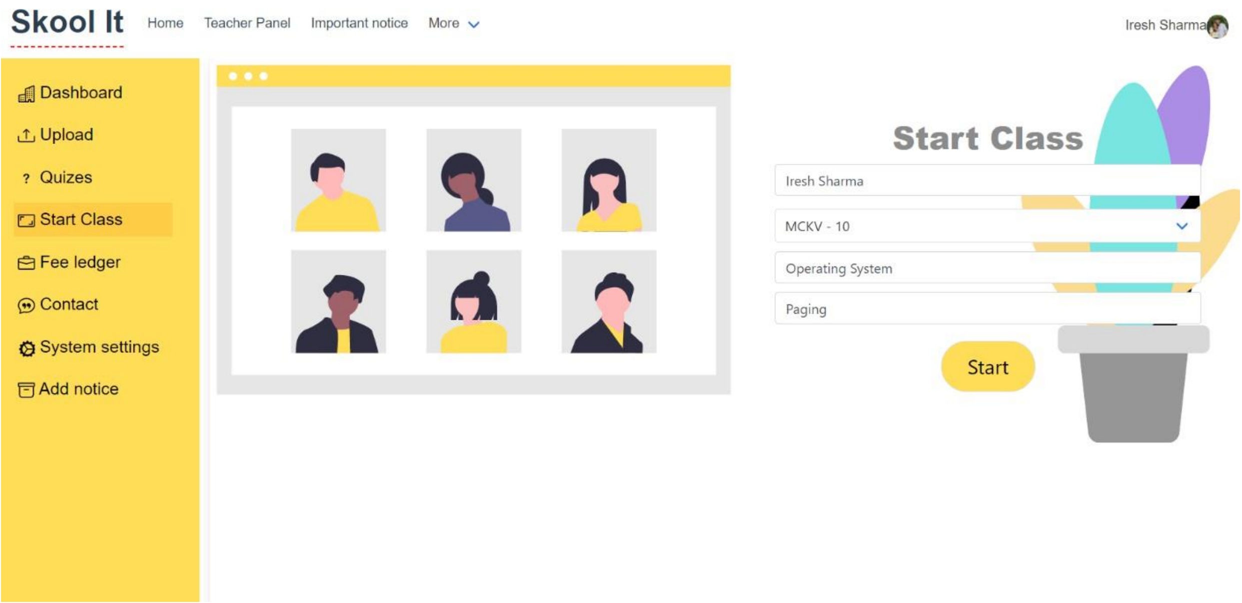
- This is where the Admin can add notes and videos of his lectures for the users.



Quiz Name	Batch Name	Actions		
try 2		View	Upload solution	Remove
quiz 1	ABBS - X	View	Upload solution	Remove

Fig. 5.13(Quizzes Option)

- The admin can start a quiz, remove it and also upload the solution of the quiz.



Iresh Sharma

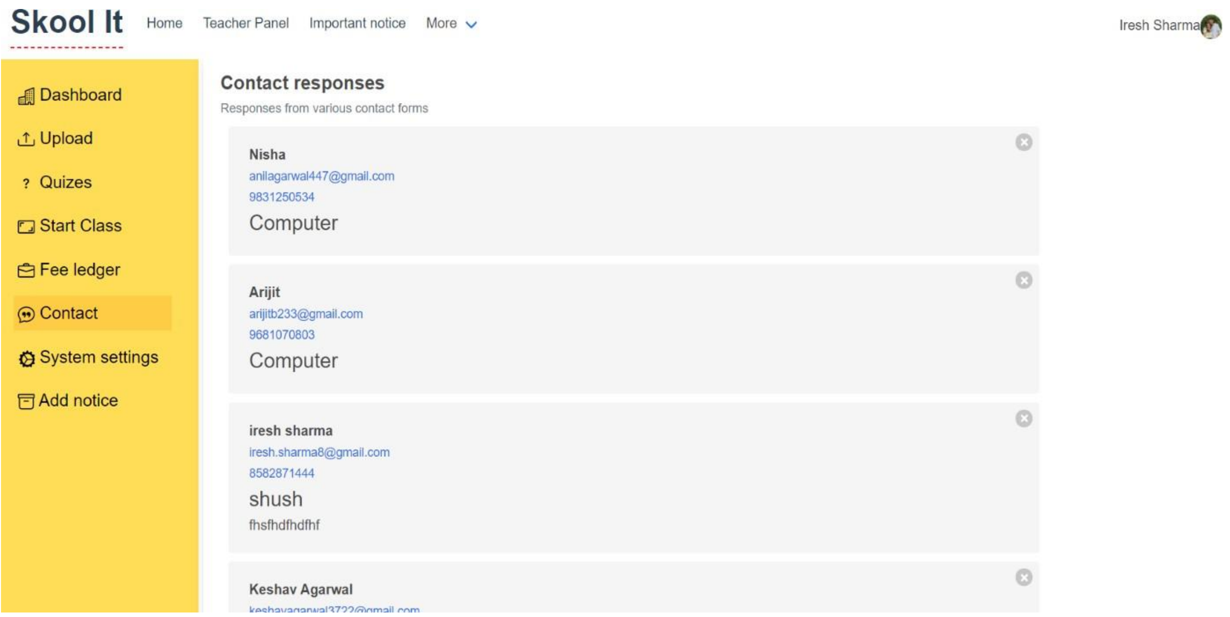
MCKV - 10

Operating System

Paging

Fig. 5.14(Start Class Option)

- The admin can start a class and conduct it batchwise for the users to make their work easier.



Contact responses

Responses from various contact forms

Nisha
anilagarwal447@gmail.com
9831250534
Computer

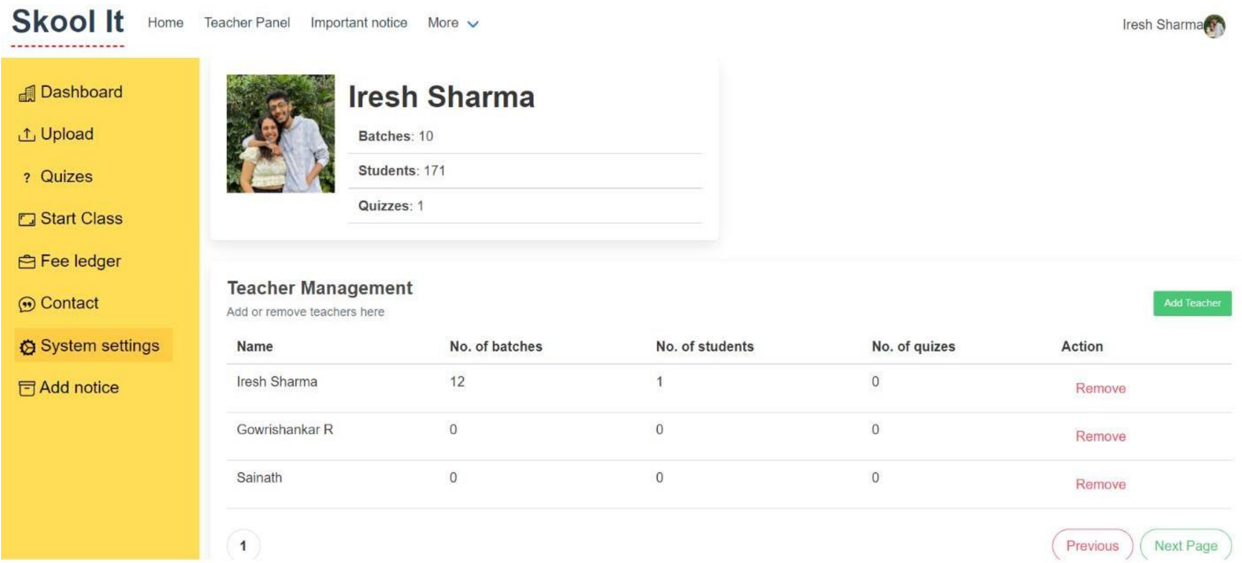
Arijit
arijitb233@gmail.com
9681070803
Computer

iresh sharma
iresh.sharma8@gmail.com
8582871444
shush
fhsfhdfhdfhf

Keshav Agarwal
keshavagarwal3722@gmail.com

Fig. 5.15(Contact Option)

- Here we can check out the contact responses from various contact forms.



Iresh Sharma

Batches: 10

Students: 171

Quizzes: 1

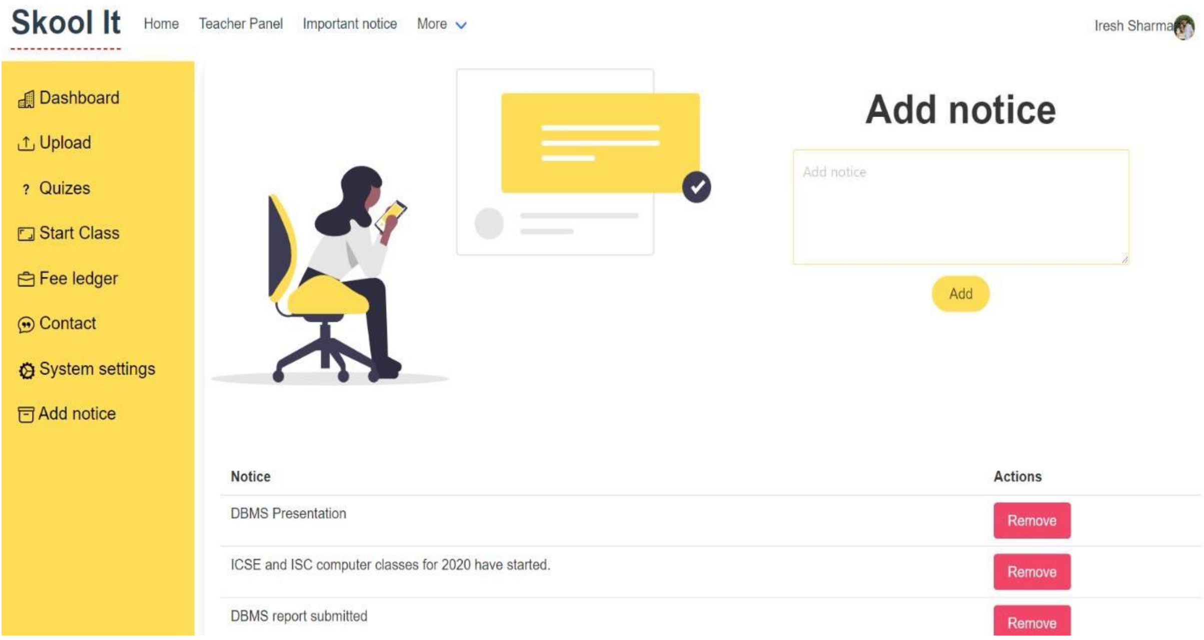
Teacher Management

Add or remove teachers here

Name	No. of batches	No. of students	No. of quizes	Action
Iresh Sharma	12	1	0	Remove
Gowrishankar R	0	0	0	Remove
Sainath	0	0	0	Remove

Fig. 5.16(System Settings Option)

- This option helps the admin to add/remove teachers who can take classes.



Add notice

Add notice

Notice	Actions
DBMS Presentation	Remove
ICSE and ISC computer classes for 2020 have started.	Remove
DBMS report submitted	Remove

Fig. 5.17(Add Notice Option)

- Here, the admin can add and also remove notices which are being displayed on the noticeboard.

Chapter 6

RESULTS

Conclusion and Future Enhancements

6.1.1 Conclusion:

In this project, we developed an automated school management system that facilitates the various activities taking place at schools. The system developed in the project consists of web applications. In all the school management system is bringing a great difference in the lives of students, teachers, parents, and the admin. Good management offers better productivity and hence more progress towards development. Seeing its demands and benefits, we have come forward with a pretty good school management system. It helps the school to achieve the target, reduce work and increase efficiency.

6.1.2 Problems Encountered:

A lot of challenges surfaced during the development of this incredible application though it tried stopping this project, but the doggedness and consistency of the writer was in match with the challenge.

The following are some of the problems or challenges encountered:

- Expensive internet facility.
- Less human power.
- Time factor on research to get a way of packaging the application successfully.

6.1.3 Future Enhancements:

- Providing login using a specific organization-id only.
- Providing sign-up using social media.
- Sharing the Notes from the website to social media apps.
- Providing star rating system to user to share their experience of the website.
- Providing a feature where the user can choose whether or not they want their resources and chats to be available publicly or remain private.
- Providing different categories of subjects for the user to label their resources, so that it can be easy for other users to sort through the items.
- Providing an option to join a club where they will receive a mail regarding the club activities.

References

- Vue JS Documentation – Vuejs.com
- Firebase Authentication Documentation – Firestack
- Firestore Documentation – Firestack
- Firebase Storage – Firestack
- HTML Information – W3Schools
- CSS Tips – W3Schools
- JavaScript Documentation – MDN
- Git and GitHub – GIT Notes