

# DBIS Project-II

M V Karthik (200010030)  
Om Patil (200010036)

30 October 2022

## Contents

<b>1</b>	<b>Overview</b>	<b>2</b>
<b>2</b>	<b>Database Design</b>	<b>3</b>
2.1	ER Model . . . . .	3
2.2	Relational Schema . . . . .	3
<b>3</b>	<b>Project Stack</b>	<b>4</b>
<b>4</b>	<b>Application Workflows</b>	<b>5</b>
4.1	Home Page . . . . .	5
4.2	Sign Up . . . . .	5
4.3	Sign In . . . . .	6
4.4	Home (for normal users) . . . . .	6
4.5	Search Course . . . . .	6
4.6	Course Planner . . . . .	7
4.7	Discussion Page . . . . .	7
4.8	Create post . . . . .	8
4.9	Discussion pagination . . . . .	8
4.10	Admin Home . . . . .	9
4.11	Add Time Slots (for admin only) . . . . .	9
4.12	Add Course (for admin only) . . . . .	10
4.13	Moderation in Discussions(for admin only) . . . . .	10
<b>5</b>	<b>Conclusions</b>	<b>11</b>

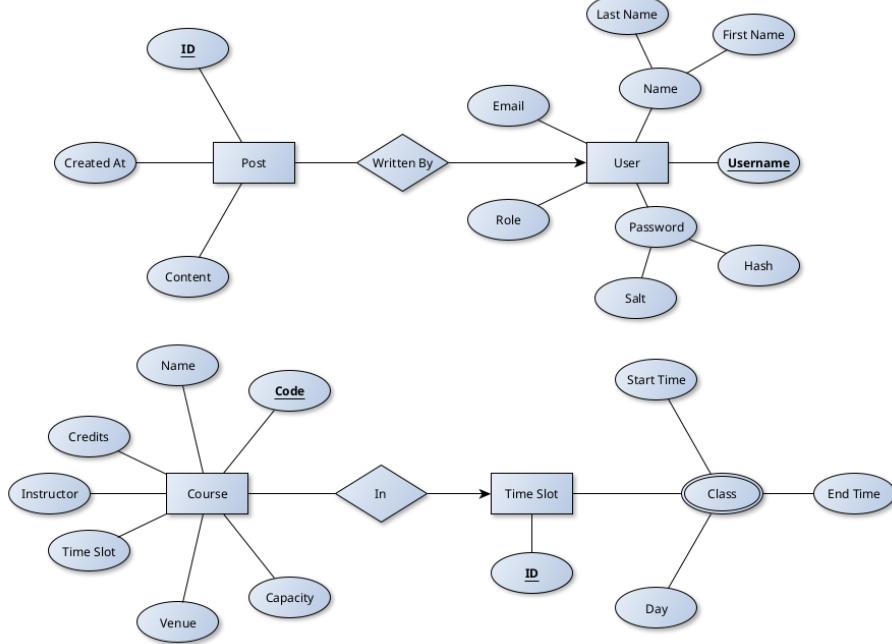
## 1 Overview

Our project is a portal designed to assist students in selecting courses during course registration. Key features include,

- Searching for available courses.
- Viewing potential timetable including selected courses.
- Detecting clashes between selected courses.
- Post forum for discussions among users.
- Easy addition of courses and slots on portal for admins.
- Discussion forum post moderation feature for admins.
- Secure account login with password salting and hashing (SHA256).
- Persistent login sessions.
- User validated inputs throughout the platform.
- Mobile friendly UI.
- Protected pages and multi-role users.

## 2 Database Design

### 2.1 ER Model



**User:** This entity contains information about the user on the portal. A user can be identified by their username. A few import attributes of users are name, password, role, etc.

**Post:** This entity contains information about posts on the discussion forum. A post can be identified by its ID. Other attributes include the time of creation and content.

**Time Slot:** This entity contains information about time slots. It has a multi-valued composite attribute class which contains the day, start and end times of classes in the time slot. A time slot can be uniquely identified by its ID.

**Course:** This entity contains information about the offered courses such as its name, code, credits, instructor, time slot, venue and capacity. A class can be uniquely indetified by its code.

### 2.2 Relational Schema

Solving this our ER model gives the following relational schema,

**Users** (username, fname, lname, email, role, salt, hash)  
**Courses** (code, name, credits, time\_slot, venue, instructor, capacity)

`Time_slots (id, day, start_time, end_time)`  
`Posts (id, author, content, created_at)`

Foreign key constraints are as follows,

`Post: author references Users`  
`Course: time_slot references Time_slots`

Other constraints are as follows,

`Users: email is unique`  
`Users: role ∈ {0, 1, 2}`  
`Users: salt is of 32 char length`  
`Users: hash is of 64 char length`  
`Time_slots: day ∈ {0, 1, ..., 6}`  
`Time_slots: start_time < end_time`

All attributes are not null except for,  
`Courses: venue, capacity`

### 3 Project Stack

Our application is designed in a three tier architecture. The technology stack for each tier is as follows,

#### Client

- Frontend: Reactjs

#### Server

- Web Server: Expressjs
- Application Server: Node.js
- Database Driver: mysqljs

#### Database

- Database: MySQL

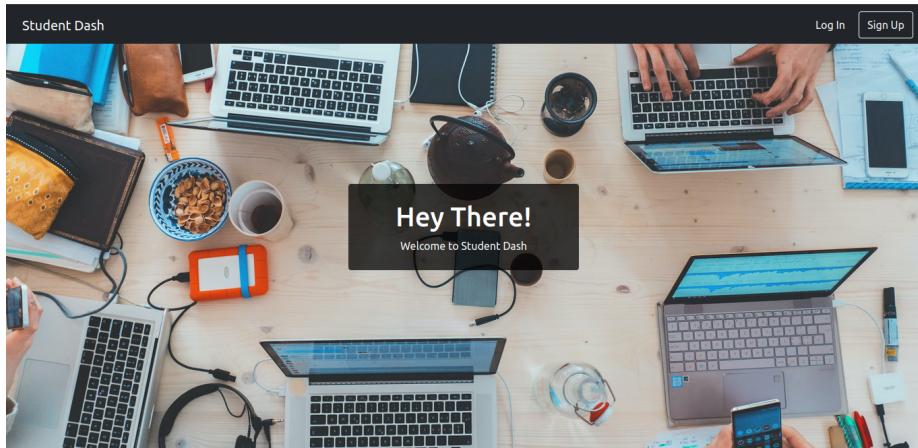
Languages used in the project include JavaScript and SQL.

As our project uses a three tier architecture the client cannot directly interact with the database hence views are not required.

## 4 Application Workflows

### 4.1 Home Page

This is the first page a user lands on when he visits the website. From here the user can go to either sign up or log in.



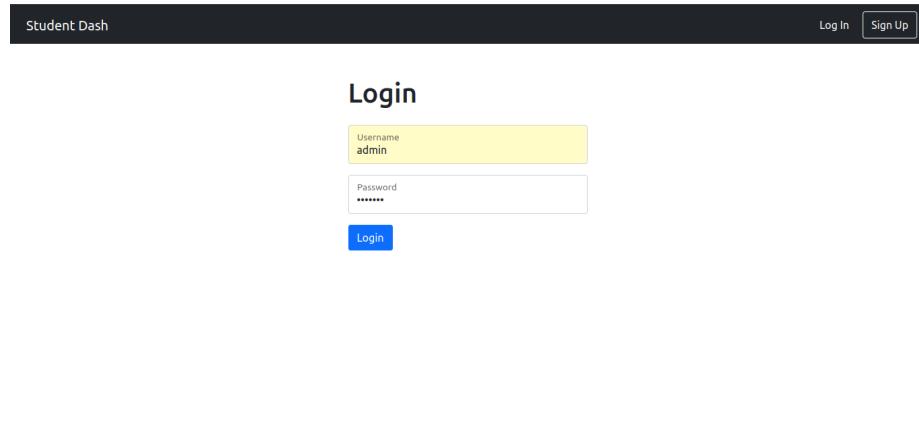
### 4.2 Sign Up

This workflow can be used to create a new account. The sign up form does basic input validation on the client side such as ensuring required fields are filled and that the password matches the confirm password. It also makes sure that the username does not already exist in the database by querying the server.

A screenshot of the 'Signup' form. The top navigation bar has 'Student Dash' on the left and 'Log In' and 'Sign Up' buttons on the right. The form itself has a title 'Signup'. It contains four input fields: 'First Name' (Karthik), 'Last Name' (Murakonda), 'Email' (abc@abc.com) which is highlighted with a green checkmark, and 'Username' (admin) which is highlighted with a red error circle and the message 'Sorry this username is taken!'. Below these are two password fields: 'Password' (\*\*\*\*\*) and 'Confirm Password' (\*\*\*\*\*), both of which are highlighted with red error circles and the message 'Passwords do not match!'. At the bottom is a blue 'Sign Up' button.

### 4.3 Sign In

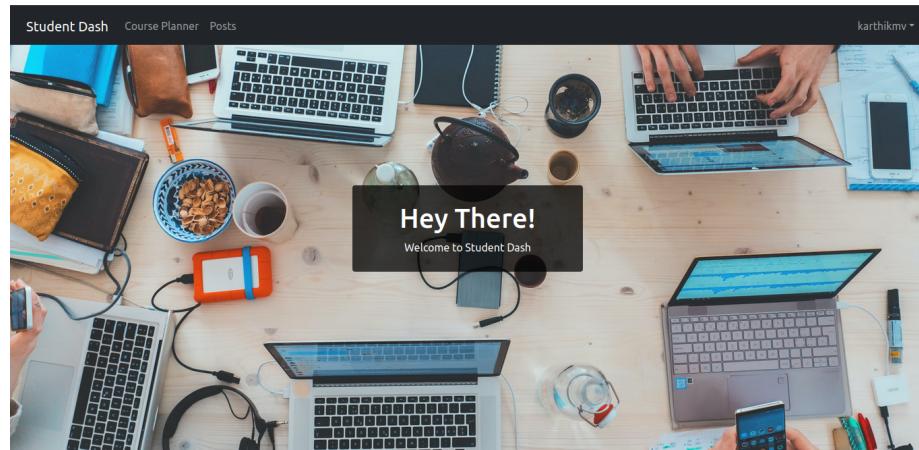
This page consists of a basic form with input validation for login. Upon successful login it goes to the home page / required page and in case of failure gives an error page.



The screenshot shows a login interface. At the top, there is a navigation bar with 'Student Dash' on the left and 'Log In' and 'Sign Up' buttons on the right. Below the navigation bar is a title 'Login'. There are two input fields: 'Username' containing 'admin' and 'Password' containing '\*\*\*\*\*'. A blue 'Login' button is located below the password field.

### 4.4 Home (for normal users)

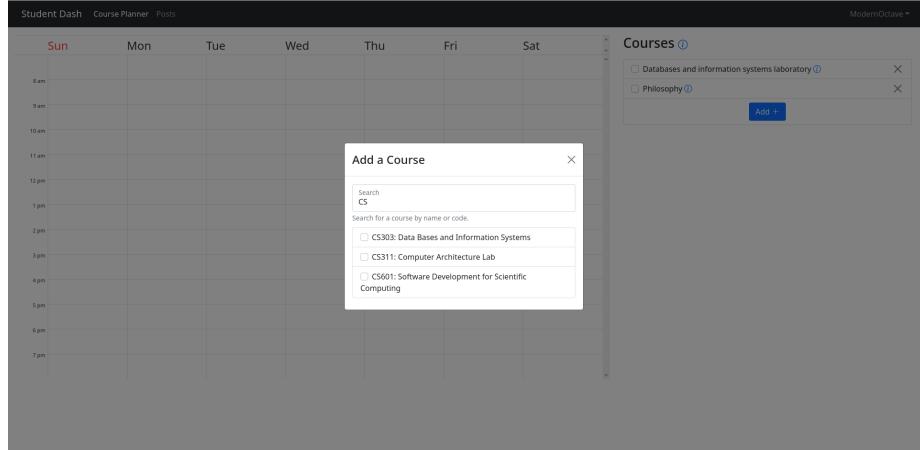
After logging in the home page now provides options to go to the course planner, discussion forum and an option to log out.



### 4.5 Search Course

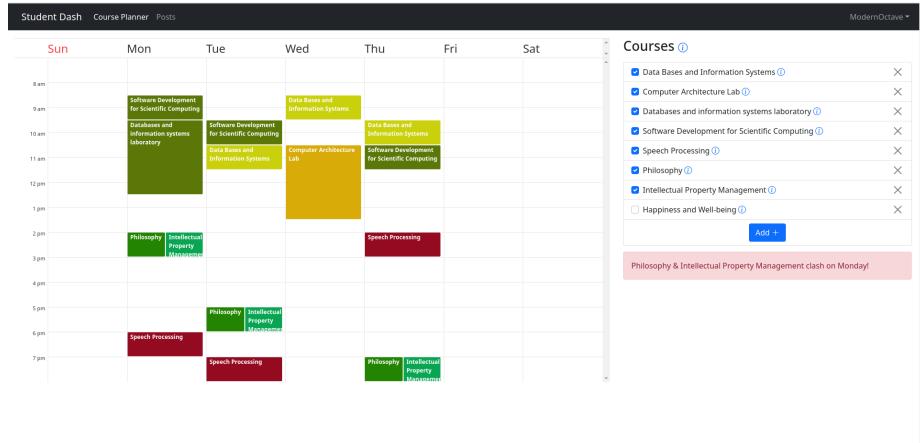
In the course planner clicking on the add button opens the search modal. Here the user can search for courses by name or code and select the ones they are

interested in.



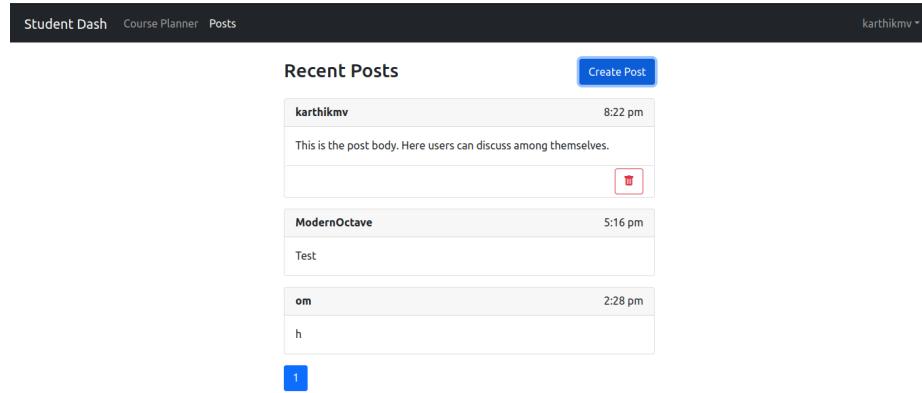
## 4.6 Course Planner

Here the user can select courses and see the potential time table. They are also informed about any time table clashes here. The add button allows them to add more courses to the list.



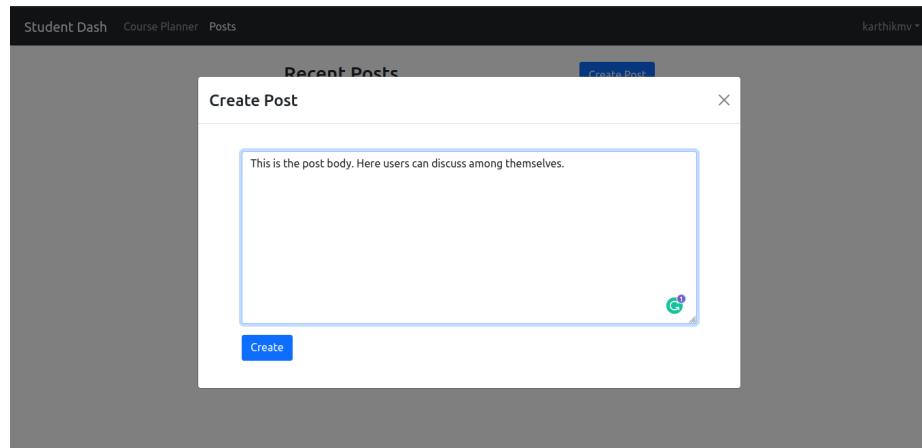
## 4.7 Discussion Page

On this page users can discuss with other users via posts. One can create a post and delete their old posts as well.



## 4.8 Create post

This simple modal allows one to type out a new post, click on the create button then publishes the post.



## 4.9 Discussion pagination

Posts are sent as paginated by backend. Image shows pagination in frontend.

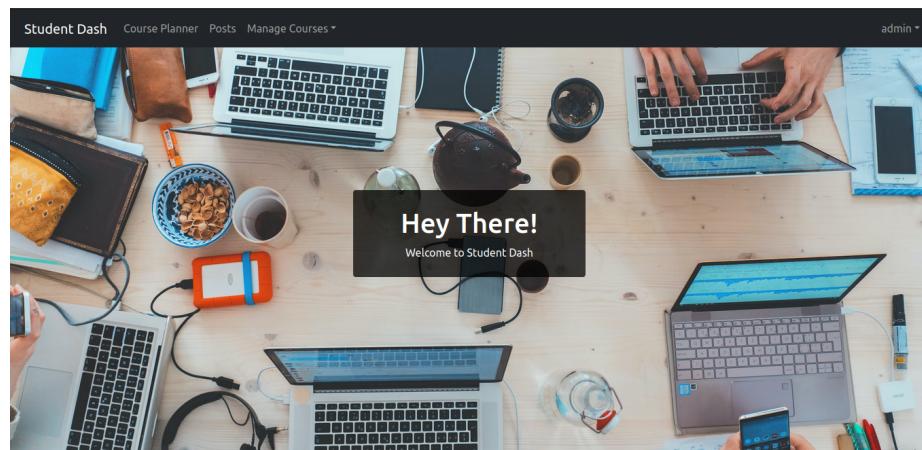
The screenshot shows a user interface for a 'Posts' section. At the top, there are navigation links: 'Student Dash', 'Course Planner', and 'Posts'. On the right, there is a user profile icon labeled 'karthikmv'. Below the header, the title 'Recent Posts' is displayed, followed by a 'Create Post' button. A table lists two posts:

User	Time
ModernOctave	5:16 pm
Test	
om	2:28 pm
h	

At the bottom of the list, there are page navigation buttons: '1' and '2'.

## 4.10 Admin Home

Admin homepage after login, notice the new option manage calendar.



## 4.11 Add Time Slots (for admin only)

Form for adding time slots on which courses will happen.



### Add a New Time Slot

Slot ID

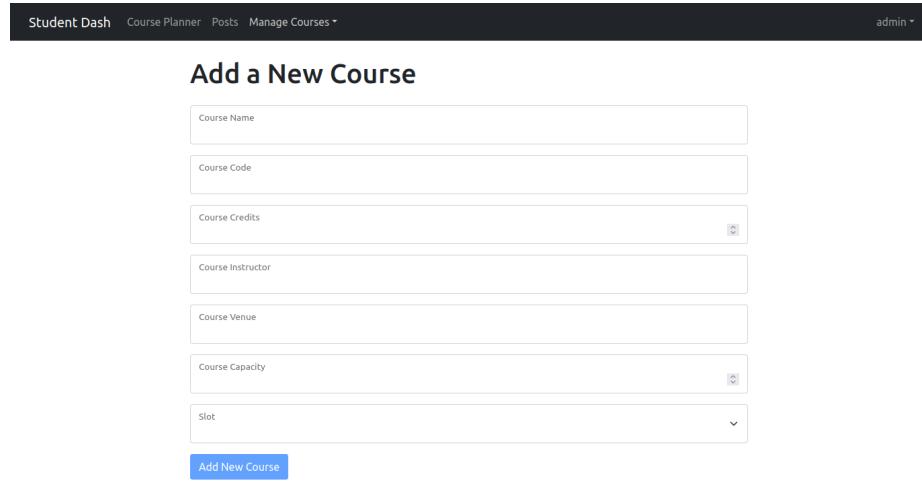
Classes

Add New Class

Add New Slot

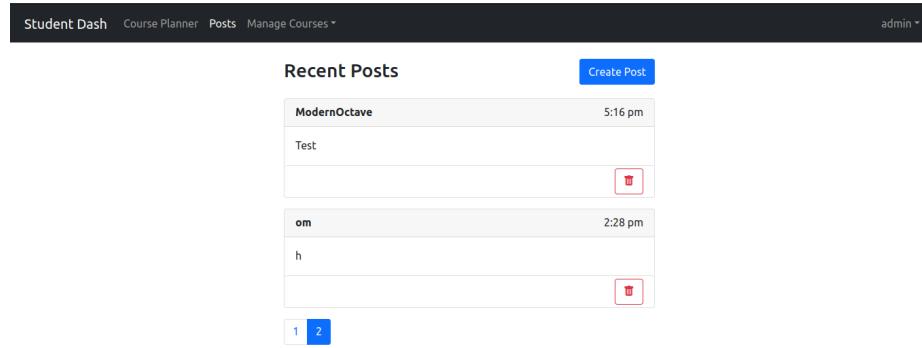
### 4.12 Add Course (for admin only)

Admin can add course with this form. He has to chose time slots from already added time slots for the new course timetable.



### 4.13 Moderation in Discussions(for admin only)

Admin has ability to delete anyone's post for sake of moderation in forum.



## 5 Conclusions

With this we have implemented the platform we set out to make. Some possible feature which we could implement in the future are,

- User Management
- Slot Management
- Course Management
- User Profiles
- Enrollment Portal

Most of these improvement are relevant to the frontend and the required back-end/database support is currently complete.