A PROJECT REPORT ON

# CERP

## College ERP

SUBMITTED IN PARTIAL

FULFILLMENT OF

DIPLOMA IN ADVANCED COMPUTING (PG-DAC)


UNDER THE GUIDANCE OF

Mr. Jaydeep Apte

PRESENTED BY

220940120013 Adarsh Gupta
220940120035 Anupama Joshi
220940120066 Disha Murarka
220940120127 Nikita Tiwari
220940120153 Rakshit Jain
220940120121 Saud Mukhair

AT

CENTER FOR DEVELOPMENT OF ADVANCED

COMPUTING C-DAC, PUNE

# ACKNOWLEDGMENT

The project “CERP” was a great learning experience for us and we are submitting this work to Advanced Computing Training School (C-DAC ACTS, Pune).

We are very glad to mention the name of Mr. Jaydeep Apte for his valuable guidance to work on this project.

We are highly grateful to Ms. Risha P. R., Manager of ACTS Training Center, CDAC, for her guidance and support whenever necessary during the course of our journey to acquire PG-Diploma in Advanced Computing (PG-DAC) through CDAC ACTS, Pune.

Our heartfelt thanks to Yogendra sir (Course Coordinator, PG-DAC) who gave us all the required support and kind coordination to provide all the necessities to complete the project and throughout the course up to the last day of the course.

From:

Adarsh Gupta
Anupama Joshi
Disha Murarka
Nikita Tiwari
Rakshit Jain
Saud Mukhair

# TABLE OF CONTENTS

# ABSTRACT

CDAC has been a major part on our career paths. As it inspired us to create this project which presents a college ERP(CERP) system that serves as a one-stop solution for students and administrators.

The ERP system includes functionalities such as student sign-up, login, profile viewing, timetable, and feedback provision for various modules. The administrator can log in, manage attendance, view feedback, generate, update, and delete schedules.

The system was developed using SPRINGBOOT and MySQL, with REACT and BOOTSTRAP for the user interface. The project's significance lies in providing a centralized system that streamlines administrative and academic processes, making them more efficient and accessible.

The system's future work could include adding more modules to cater to other academic requirements and expanding the user base.

# INTRODUCTION

The purpose of this project is to develop a college ERP system that caters to the needs of students and administrators. As the number of students and courses offered by colleges/school or any educational institution increases, there is a need for a centralized system that can streamline administrative and academic processes.
Our college ERP system aims to achieve this by providing a single platform for all academic and administrative activities..

The system includes functionalities such as student sign-up, login, profile viewing, timetable, results, and feedback provision for various modules. The administrator can log in, manage attendance, view feedback, generate, update, and delete schedules. The system was developed using SPRINGBOOT and MySQL, with REACT and BOOTSTRAP for the user interface. Our project aims to provide an efficient and easy-to-use system that simplifies the process of managing academic and administrative activities.

In this report, we will discuss the various aspects of our college ERP system, including the development process, technical details, testing and evaluation, and the results of the system's implementation. We will also explore the potential applications of the system and the future work that could be done to improve it.

Overall, this project report aims to showcase the effectiveness of our college ERP system and its potential to transform administrative and academic processes in different educational institutions.

# PROJECT OVERVIEW AND SUMMARY

## 3.1. PURPOSE

Our project, “CERP”, is a web-based online college management portal which aims to provide admin with functionalities to manage students efficiently. The purpose of this project is to develop a college ERP system that caters to the needs of students and administrators.

## 3.2. SCOPE

“CERP” aims to deliver a web-based portal that manages all the students where Course-Coordinator or class Teacher can be the Admin. Students can sign up and view their profile, get schedule, view their attendance in attractive graphical manner, get the results and also give their feedback for each modules.
Admins can login and add or update attendance for registered students, upload results, get feedback list, generate schedule and update them as well.

CERP is a portal for both students (for course related activities)
and admins (for managing students).

## 3.3. OVERVIEW

### A. TECHNOLOGIES USED

I. FRONT END

- HTML
- CSS
- Bootstrap
- React Js
- React-Redux
- JavaScript
- Jquery

II. BACK END

- Spring Boot
- Spring Data JPA
- Spring Security
- Spring Restful Services

III. DATABASE MANAGEMENT SYSTEM

- MySQL

## *B. FEATURES PROVIDED*

I. FOR STUDENTS

1. **Register:** Students can register themselves by adding general details about themselves.

2. **Login:** Successfully registered candidates are now eligible to Login using the user ID and password they have set while signing up.

3. **View Profile**: After successful login, students can view their profile.

4. **Give Feedback:** Students can give feedback for each module based on provided criteria.

5. **View Time-Table:** Students can view the schedule day wise / month-wise / Year wise or even on weekly basis.

6. **View Attendance:** Students can view their overall attendance in different module in attractive graphical representation.

7. **Logout** – After utilizing the portal, students can Logout of the portal.

## II. FOR ADMINS

**1. Login & Logout:** Similar to students, admins can login & logout to access their account.

**2. Add Attendance :** For successfully registered students admin will be able to add their attendance. First admin needs to select the course for which students has enrolled. Then Admin need to choose the subject for which they wish to log the attendance. Now the whole list of students will appear on the screen and Admin can log the attendance for each student.

**3. Update Attendance:** Admin can also update the previously logged attendance in similar manner as he added the attendance.

**4. Generate Schedule:** Admin can generate the schedule for whole month. As soon as Admin click on this option , a calendar will appear on the screen wherein each cell represents a date and by clicking on that date , Admin will be given a form wherein he can edit enter the details like start date/time, end date/time and subject etc.

**5. Update Schedule:** Admin can update the schedule as he added the event.

**6. Delete Schedule:** Admin can also delete any event from the schedule.

**7. View Feedback List:** Admin can view the feedback given by students for each module collectively.

**8. Log out:** Admin can logout after using the portal.

## *3.4. FEASIBILITY STUDY*

Feasibility is the determination of whether a project is worth undertaking or not. Before actually recommending the new system, it is important to investigate if it is feasible to develop it.

Before developing and implementing a system, we have to make sure that the system is feasible in the following ways:

A. Technical Feasibility
B. Operational Feasibility
C. Economical Feasibility

## *A. TECHNICAL FEASIBILITY*

In this type of feasibility study, the system analyst has to check whether it is possible or not to develop the requested system with the available manpower, software, hardware, etc.

This project makes use of cross-platform software and solutions like Java, and hence can run on any operating system. JavaScript, used in front-end, is swift and versatile framework when it comes to delivering the requested page. Also, as JavaScript is popular, it is easy to learn it and utilizing it as front end technology. The combination of Spring Boot, Spring Data JPA and Spring Security for backend make for a fast, easy to set-up and reliable system to interact with the database, as they are secure and transactional in nature. Since the sensitive data of customers and admins need to be stored in a robust and secure database, MySQL database management system was chosen as it is an industry standard.

## *B. OPERATIONAL FEASIBILITY*

In this type of feasibility study, the operation of the system is considered. An analysis is performed on whether it is feasible for the user department to use the application. Thus, the proposed system is said to be operationally feasible only if clients are able to understand the system clearly and correctly, and can use it wIn the design of this project, we always kept user experience in mind. We made an effort to have a good user interface with consistent theme and alluring design to keep the users interested and engaged. In our project, the use of universally known icons and instructions that are easy to understand makes sure that the user will not need any special technical know-how to use the application. We made sure that the information available throughout the application is arranged in a logically coherent and consistent manner, guaranteeing that the users will have a smooth and effortless experience and even enjoy using the application.

## *C. ECONOMIC FEASIBILITY*

In this type of feasibility study, the benefits of the system to the organization are considered by taking into consideration the cost-benefit analysis. Almost all the software and technologies used in our project is free, open-source, and widely available, with each of the technologies having an extensive community support. This makes “CERP” an economically feasible solution to the organizations that wish to implement it.

# REQUIREMENTS-FULFILLED

## *4.1. FUNCTIONAL REQUIREMENTS*

Following are the functional requirements fulfilled by our project:

- Students can register themselves if they have completed their admission formalities successfully.
- Further, they are asked to fill in their personal details and address details.
- Students can login using their user id and password they set while signing up.
- Students now can view their profile, check their attendance and schedule for the month/week/day.
- Students can also give feedback for different module after successful completion of each module.
- Admins can Add/Update/Delete time table.
- Admin can also view the feedback given by the students in collective manner.
- Admins can Add or Update the attendance for all the students of different courses.

## *4.2. NON-FUNCTIONAL REQUIREMENTS*

Following are the non-functional requirements fulfilled by our project:

- Since the application uses lightweight and established software components that are also cross-platform, it shows remarkably good performance and has good support for every operating system.
- The use of JavaScript and Jquery for front end and Spring Boot, Spring Data JPA and Spring Security for back end delivers quick response times to admins and students.
- Card-style UI and well-known icons and symbols used throughout the application provides a consistent theme and user-friendly interface that anyone can grasp easily, even without a technical background.

# PROJECT DESIGN

## 5.1. DATA MODEL

The following tables depict the database design used for “CERP” application:

### A. Tables In CERP

```
mysql> use cerp;
Database changed
mysql> show tables;
+----------------+
| Tables_in_cerp |
+----------------+
| admin          |
| attendance     |
| course         |
| feedback       |
| schedule       |
| student        |
| subject        |
+----------------+
7 rows in set (0.03 sec)

mysql>
```

# B. Student Table

```
mysql> select * from student;
+----+-----------+---------------------------+------------+--------+-----------+---------------+-----------+
| id | address   | email                     | first_name | gender | last_name | password      | course_id |
+----+-----------+---------------------------+------------+--------+-----------+---------------+-----------+
|  1 | Pune      | dishamurarka1117@gmail.com | Disha      | Female | Murarka   | Dish@111      |         1 |
|  2 | Pune      | rakshitjain@gmail.com      | Rakshit    | Male   | Jain      | Rakshit@111   |         1 |
|  3 | Mumbai    | saud@gmail.com             | Saud       | Male   | Mukhair   | Saud@789      |         1 |
|  4 | Nashik    | anu@gmail.com              | Anu        | Female | Joshi     | Anu@789       |         1 |
|  5 | Agra      | mahak@gmail.com            | Mahak      | Female | Jain      | Mahak@789     |         2 |
|  6 | Nashik    | sneha@gmail.com            | Sneha      | Female | Choudhari | Sneha@789     |         2 |
|  7 | Gondia    | paras@gmail.com            | Paras      | Male   | Kesare    | Paras@789     |         2 |
|  8 | Amravati  | jitendra@gmail.com         | Jitendra   | Male   | Rajput    | Jitendra@789  |         2 |
|  9 | Delhi     | nikhil@gmail.com           | Nikhil     | Male   | Gupta     | nikhil@789    |         2 |
| 10 | Nimaj     | ashi@gmail.com             | Ashi       | Female | Jain      | Ashi@789      |         1 |
| 11 | Surat     | nipun@gmail.com            | Nipun      | Male   | Taunk     | Nipun@789     |         1 |
| 12 | Gwalior   | aman@gmail.com             | Aman       | Male   | Bhadoriya | Aman@789      |         1 |
| 13 | Allahabad | abhinav@gmail.com          | Abhinav    | Male   | Shukla    | Abhinav@789   |         1 |
| 14 | Indore    | yash@gmail.com             | Yashvardhan | Male  | Chauhan   | Yash@789      |         1 |
| 15 | Pune      | pratik@gmail.com           | Pratik     | Male   | Agrawal   | Pratik@789    |         1 |
| 16 | Allahabad | deepesh@gmail.com          | Deepesh    | Male   | Kuwar     | Deepesh@789   |         1 |
| 17 | Nagpur    | kajal@gmail.com            | Kajal      | Female | Khidalkar | Kajal@789     |         1 |
| 18 | Delhi     | divyangi@gmail.com         | Divyangi   | Female | Aggarwal  | Divyangi@789  |         1 |
| 19 | Pune      | kajalk@gmail.com           | Kajal      | Female | Kesare    | Kajalk@789    |         2 |
| 20 | Pune      | chaitrali@gmail.com        | Chaitrali  | Female | Shinde    | Chaitrali@789 |         2 |
| 21 | Ratnagiri | juilee@gmail.com           | Juilee     | Male   | Bhave     | Juilee@789    |         2 |
| 22 | Dhule     | harshada@gmail.com         | Harshada   | Female | Choudhari | Harshada@789  |         2 |
| 23 | Islampur  | aishwarya@gmail.com        | Aishwarya  | Female | Dabhole   | Aishwarya@789 |         2 |
| 24 | Amravati  | madhav@gmail.com           | Madhav     | Male   | Mundhada  | Madhav@789    |         2 |
| 25 | Amravati  | utkarsh@gmail.com          | Utkarsh    | Female | Deshmukh  | Utkarsh@789   |         2 |
| 26 | Amravati  | mrunmayee@gmail.com        | Mrunmayee  | Female | Chaudhari | Mrunmayee@789 |         2 |
| 27 | Nagpur    | dnyanvi@gmail.com          | Dnyanvi    | Female | Bante     | Dnyanvi@789   |         2 |
| 28 | Pune      | jayesh@gmail.com           | Jayesh     | Male   | Jaiswal   | Jayesh@789    |         2 |
| 29 | Aurangabad | kanak@gmail.com           | Kanak      | Female | Murarka   | Kanak@789     |         1 |
| 30 | Aurangabad | tanmay@gmail.com          | Tanmay     | Male   | Bagdiya   | Tanmay@789    |         1 |
+----+-----------+---------------------------+------------+--------+-----------+---------------+-----------+
30 rows in set (0.00 sec)
```

# C. Admin Table

```
mysql> select *from admin ;
+----+-------------------+------------+
| id | email             | password   |
+----+-------------------+------------+
|  1 | admin@gmail.com   | Admin@123  |
|  2 | adarsh@gmail.com  | Adarsh@789 |
+----+-------------------+------------+
2 rows in set (0.00 sec)
```

## D. Course Table

```
mysql> select * from course;
+----+-------------+
| id | course_name |
+----+-------------+
|  1 | PG-DAC      |
|  2 | PG-DBDA     |
+----+-------------+
2 rows in set (0.00 sec)
```

## E. Subject Table

```
mysql> select * from subject ;
+----+-----------------------+-----------+
| id | subject_name          | course_id |
+----+-----------------------+-----------+
|  1 | Core Java             |         1 |
|  2 | DSA                   |         1 |
|  3 | DBT                   |         1 |
|  4 | WPT                   |         1 |
|  5 | OS                    |         1 |
|  6 | DOTNET                |         1 |
|  7 | J2EE                  |         1 |
|  8 | SDM                   |         1 |
|  9 | Linux Programming     |         2 |
| 10 | DBMS                  |         2 |
| 11 | Python                |         2 |
| 12 | OOP Java              |         2 |
| 13 | Big Data Technologies |         2 |
| 14 | Data Visualization    |         2 |
| 15 | Statistics            |         2 |
| 16 | Machine Learning      |         2 |
+----+-----------------------+-----------+
16 rows in set (0.00 sec)
```

# F. Attendance Table

```
mysql> select * from attendance;
+-----+------------+------------+------------+
| id  | attendance | student_id | subject_id |
+-----+------------+------------+------------+
|   1 |         80 |          1 |          1 |
|   2 |         90 |          1 |          2 |
|   3 |         85 |          1 |          3 |
|   4 |         70 |          1 |          4 |
|   5 |        100 |          1 |          5 |
|   6 |         80 |          1 |          6 |
|   7 |         75 |          1 |          7 |
|   8 |         60 |          1 |          8 |
|   9 |         60 |          2 |          1 |
|  10 |         75 |          2 |          2 |
|  11 |         85 |          2 |          3 |
|  12 |         70 |          2 |          4 |
|  13 |         99 |          2 |          5 |
|  14 |         80 |          2 |          6 |
|  15 |         76 |          2 |          7 |
|  16 |         95 |          2 |          8 |
|  17 |         70 |          3 |          1 |
|  18 |         70 |          3 |          2 |
|  19 |         85 |          3 |          3 |
|  20 |         60 |          3 |          4 |
|  21 |         76 |          3 |          5 |
|  22 |         81 |          3 |          6 |
|  23 |         77 |          3 |          7 |
|  24 |         97 |          3 |          8 |
|  25 |         90 |          4 |          1 |
|  26 |        100 |          4 |          2 |
|  27 |         90 |          4 |          3 |
|  28 |         99 |          4 |          4 |
|  29 |         87 |          4 |          5 |
```

# G. Schedule Table

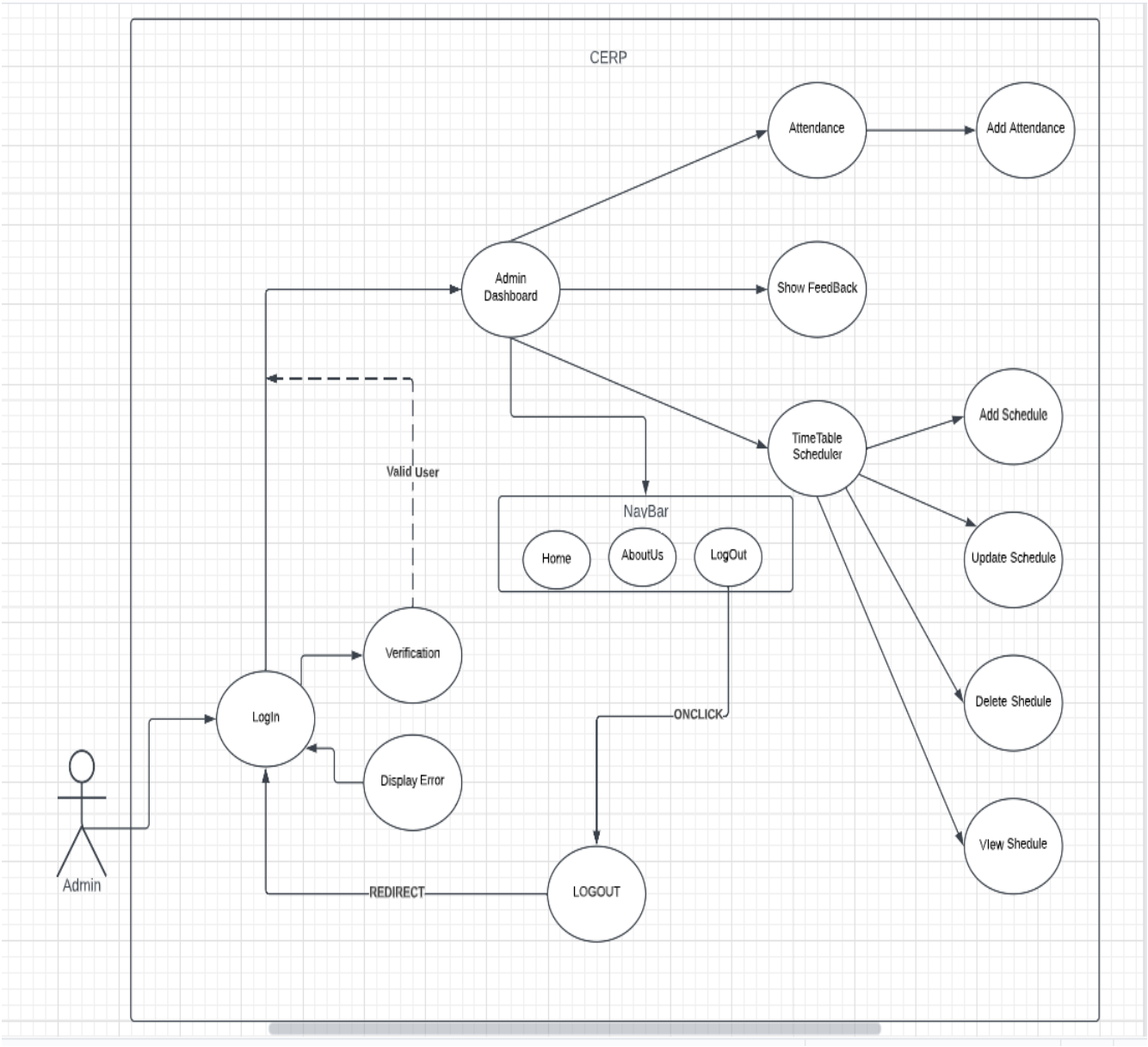
```
mysql> select * from schedule;
+----+---------------------------+----------+---------------------------+----------+------------+
| id | end_time                  | location | start_time                | course_id | subject_id |
+----+---------------------------+----------+---------------------------+----------+------------+
|  1 | 2023-03-14 11:30:00.000000 | Class 001 | 2023-03-14 08:00:00.000000 |         1 |          1 |
|  2 | 2023-03-14 16:00:00.000000 | Class 001 | 2023-03-14 13:00:00.000000 |         1 |          8 |
|  3 | 2023-03-15 12:00:00.000000 | Class 001 | 2023-03-15 09:00:00.000000 |         1 |          4 |
|  4 | 2023-03-15 21:30:00.000000 | Class 001 | 2023-03-15 18:00:00.000000 |         1 |          2 |
|  5 | 2023-03-16 13:00:00.000000 | Class 001 | 2023-03-16 10:00:00.000000 |         1 |          5 |
|  6 | 2023-03-16 18:00:00.000000 | Class 001 | 2023-03-16 15:00:00.000000 |         1 |          7 |
|  7 | 2023-03-14 13:00:00.000000 | Lab 102   | 2023-03-14 10:00:00.000000 |         2 |         11 |
|  8 | 2023-03-14 12:00:00.000000 | Lab 102   | 2023-03-14 08:00:00.000000 |         2 |         12 |
|  9 | 2023-03-16 13:30:00.000000 | Lab 102   | 2023-03-16 09:30:00.000000 |         2 |          9 |
| 10 | 2023-03-16 16:00:00.000000 | Lab 102   | 2023-03-16 13:00:00.000000 |         2 |         15 |
| 11 | 2023-03-17 20:00:00.000000 | Lab 102   | 2023-03-17 18:00:00.000000 |         2 |         12 |
+----+---------------------------+----------+---------------------------+----------+------------+
11 rows in set (0.00 sec)
```

# H. Feedback Table

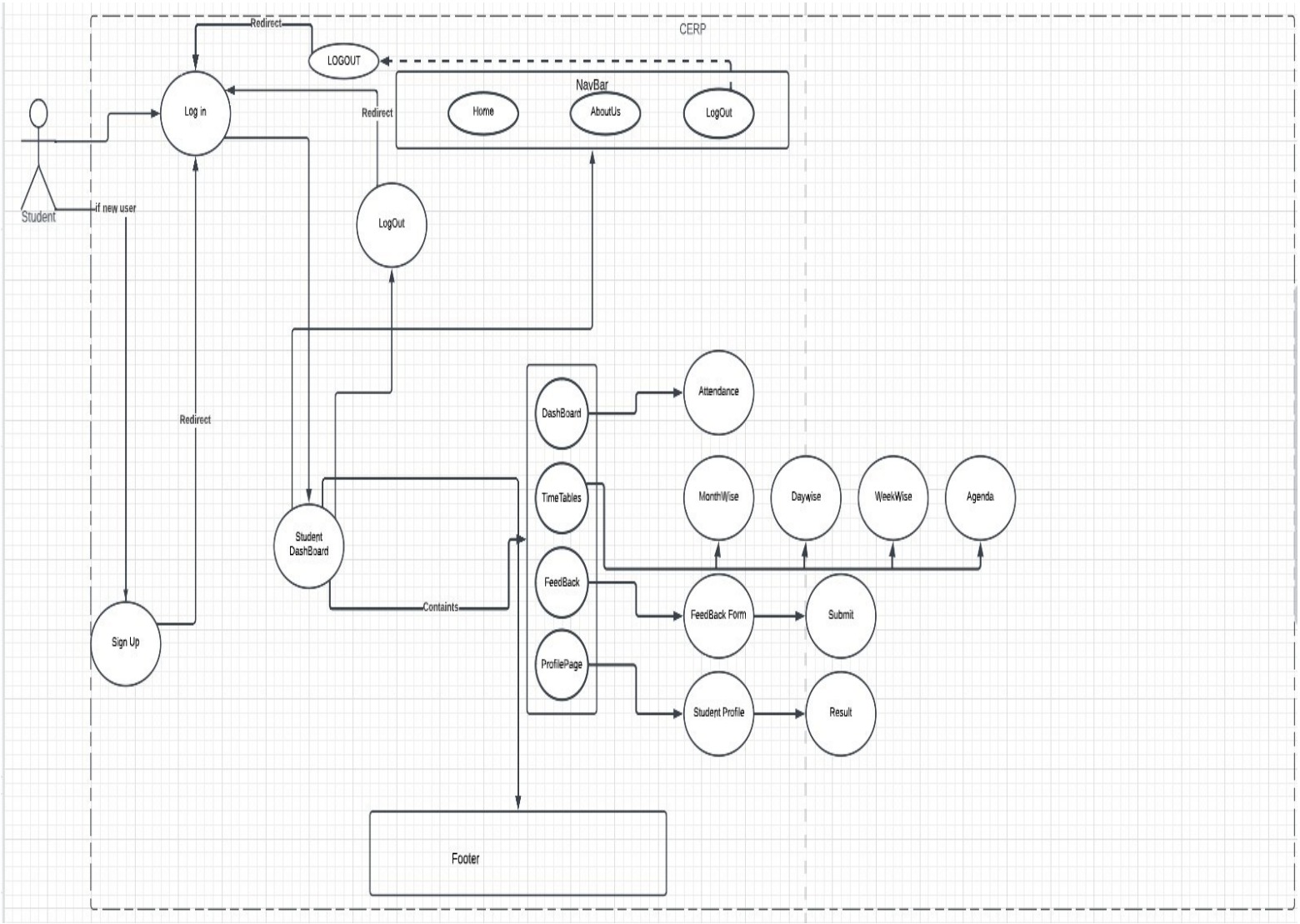
```
mysql> desc feedback;
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| id            | bigint       | NO   | PRI | NULL    | auto_increment |
| communication | int          | NO   |     | NULL    |                |
| guidance      | int          | NO   |     | NULL    |                |
| knowledge     | int          | NO   |     | NULL    |                |
| punctuality   | int          | NO   |     | NULL    |                |
| suggestion    | varchar(255) | YES  |     | NULL    |                |
| teaching      | int          | NO   |     | NULL    |                |
| student_id    | bigint       | YES  | MUL | NULL    |                |
| subject_id    | bigint       | YES  | MUL | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
9 rows in set (1.83 sec)
```

## 5.2. USE CASE DIAGRAM

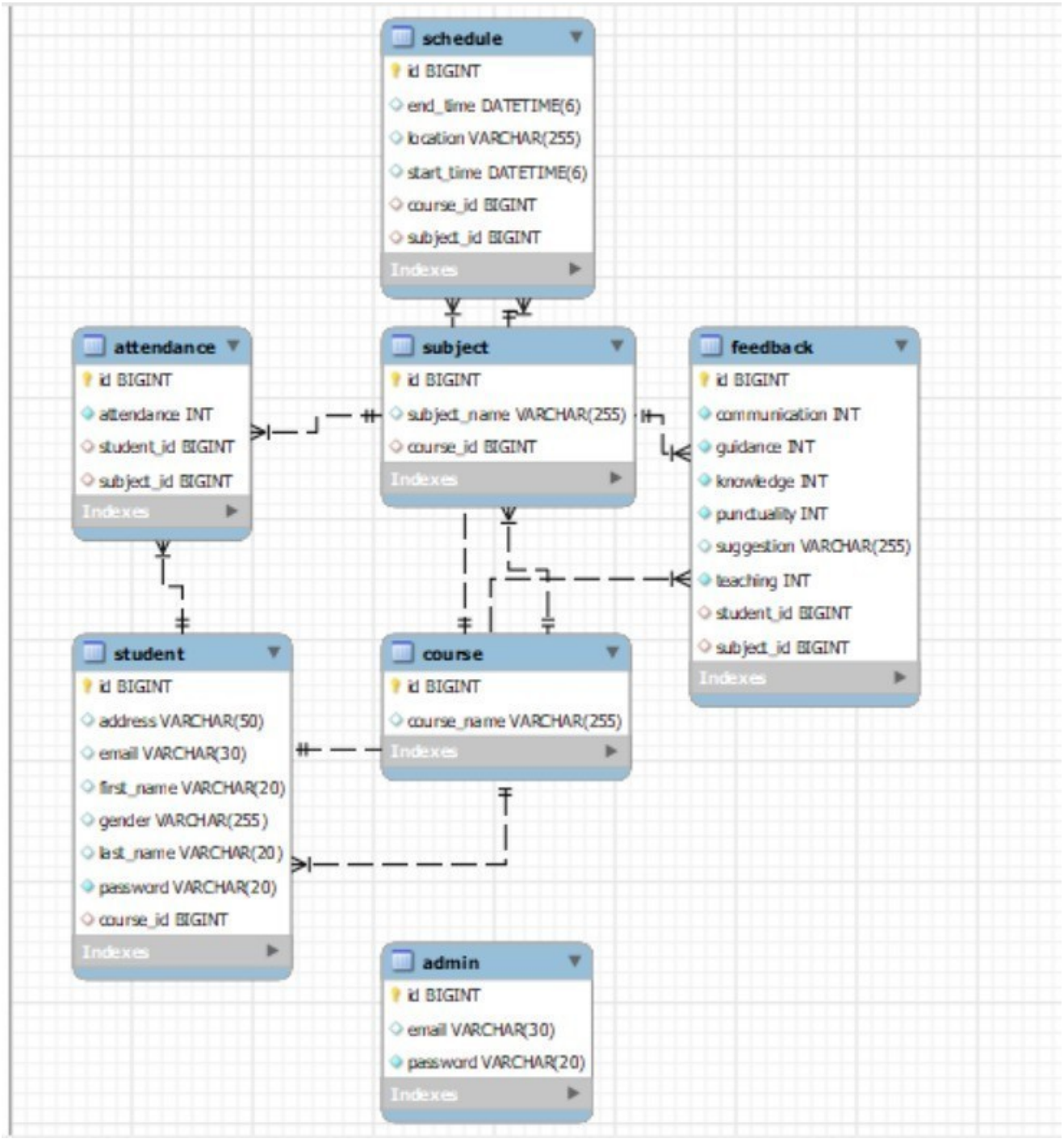
## A. ADMIN



# B. Student


CERP
Redirect
LOGOUT
Log in
Redirect
NavBar
Home
AboutUs
LogOut
Student
if new user
LogOut
Redirect
DashBoard
Attendance
TimeTables
MonthWise
Daywise
WeekWise
Agenda
Student DashBoard
FeedBack
FeedBack Form
Submit
Containts
Sign Up
ProfilePage
Student Profile
Result
Footer

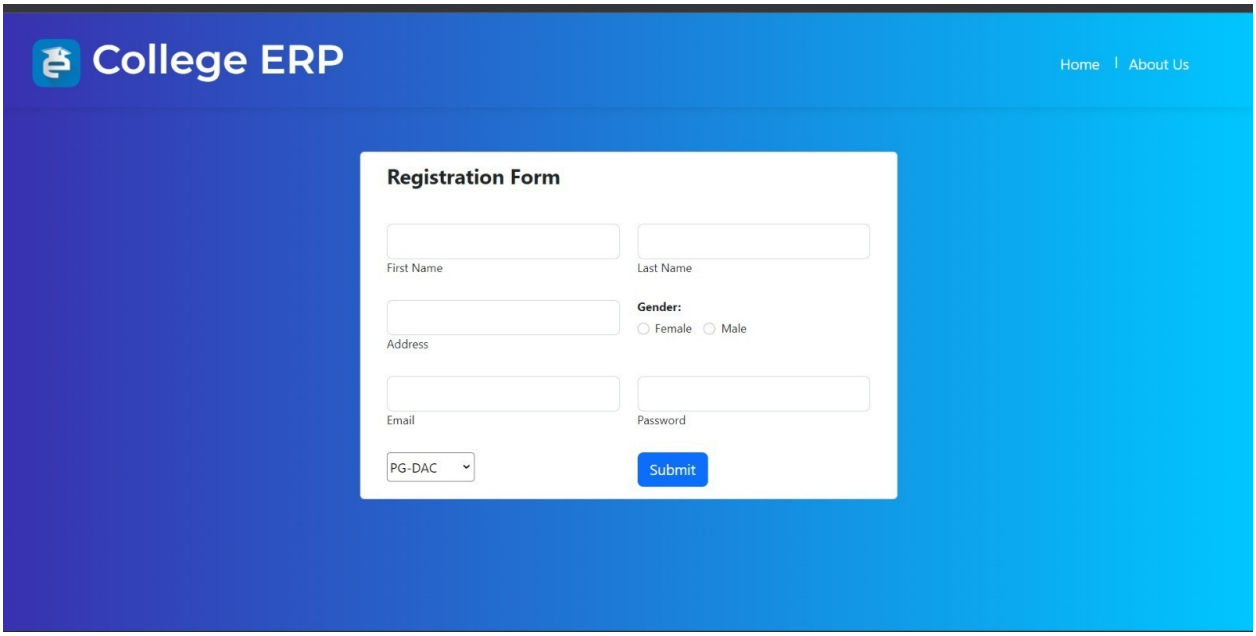
## 5.3. ER-DIAGRAM



# PROJECT SCREENSHOTS

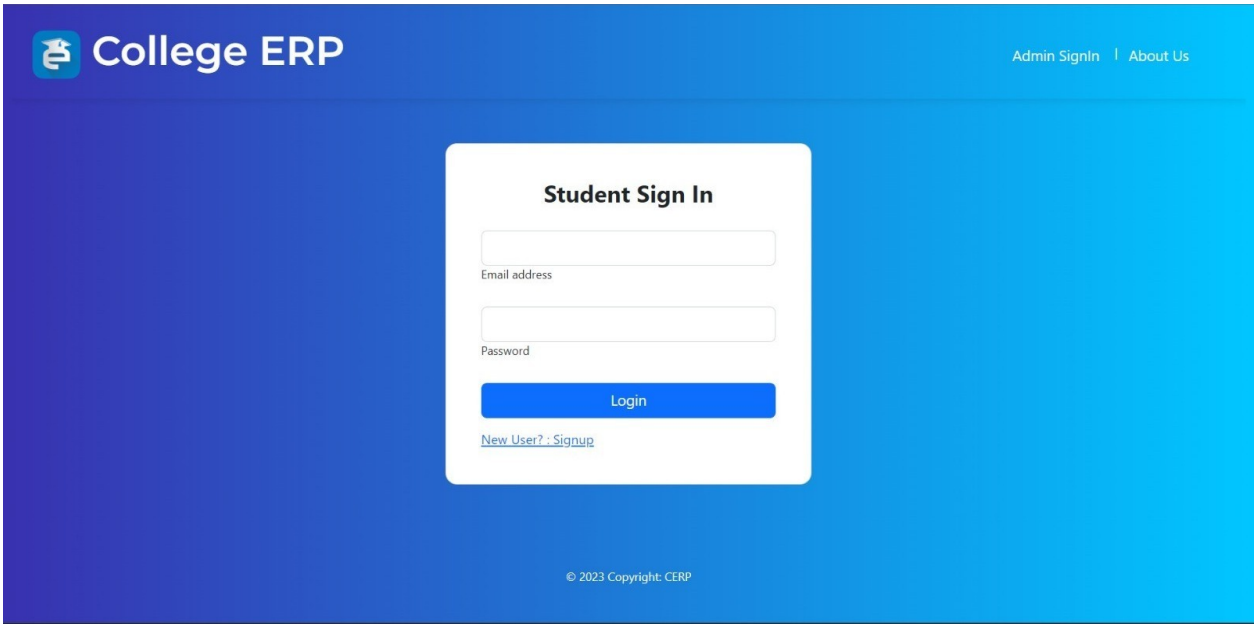
## 6.1. STUDENT

Register

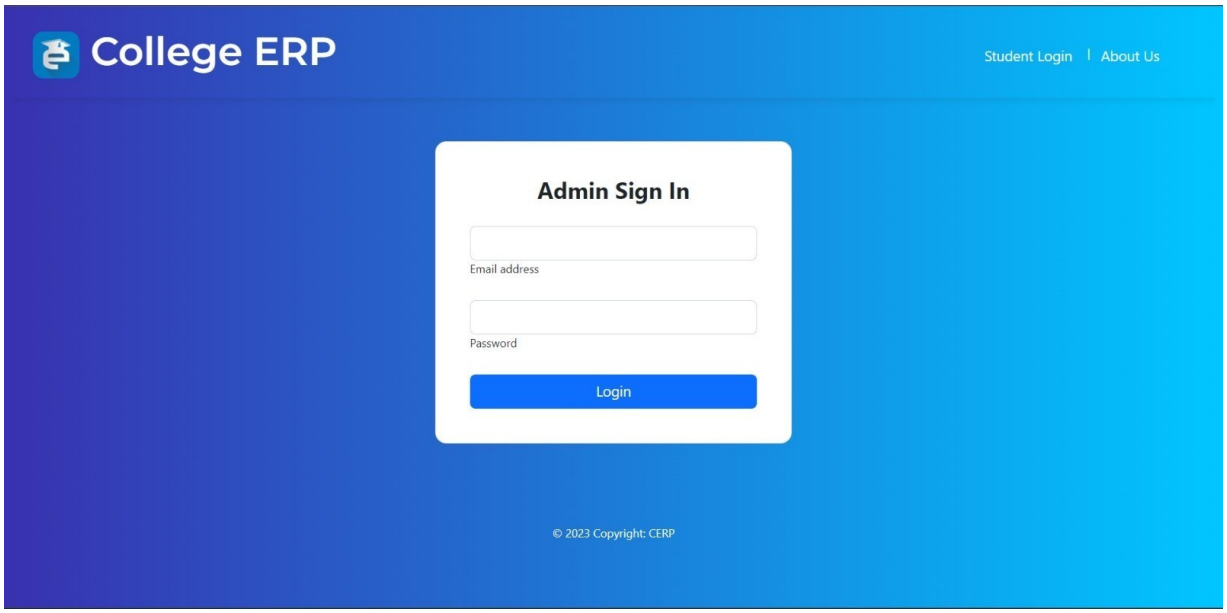



- Here they are asked to fill their personal details.
- Once the user has signed up then he wont be able to sign up again with same email address.
- Email address should be a valid email address eg xyz@abc.com
- Password should be of minimum 6 characters.
- Password should consist atleast one special character and one numeric character.

-After filling personal details , student sign up is completed and now Student can LOGIN via correct credentials.

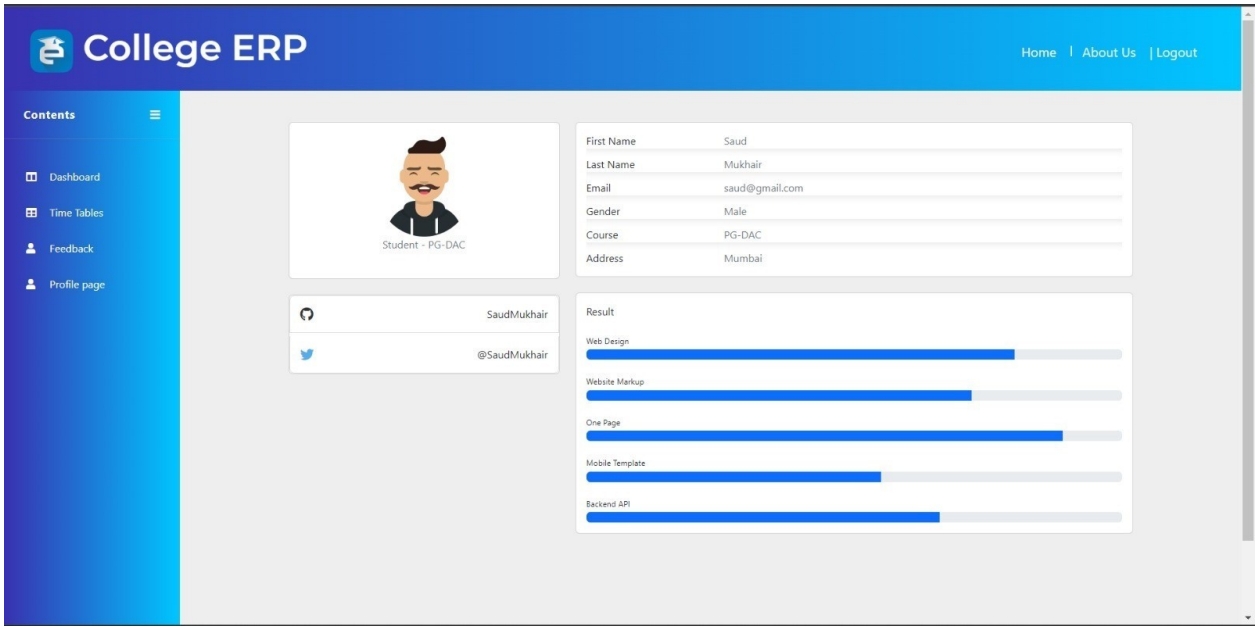



-Admin can also sign in using correct credentials.

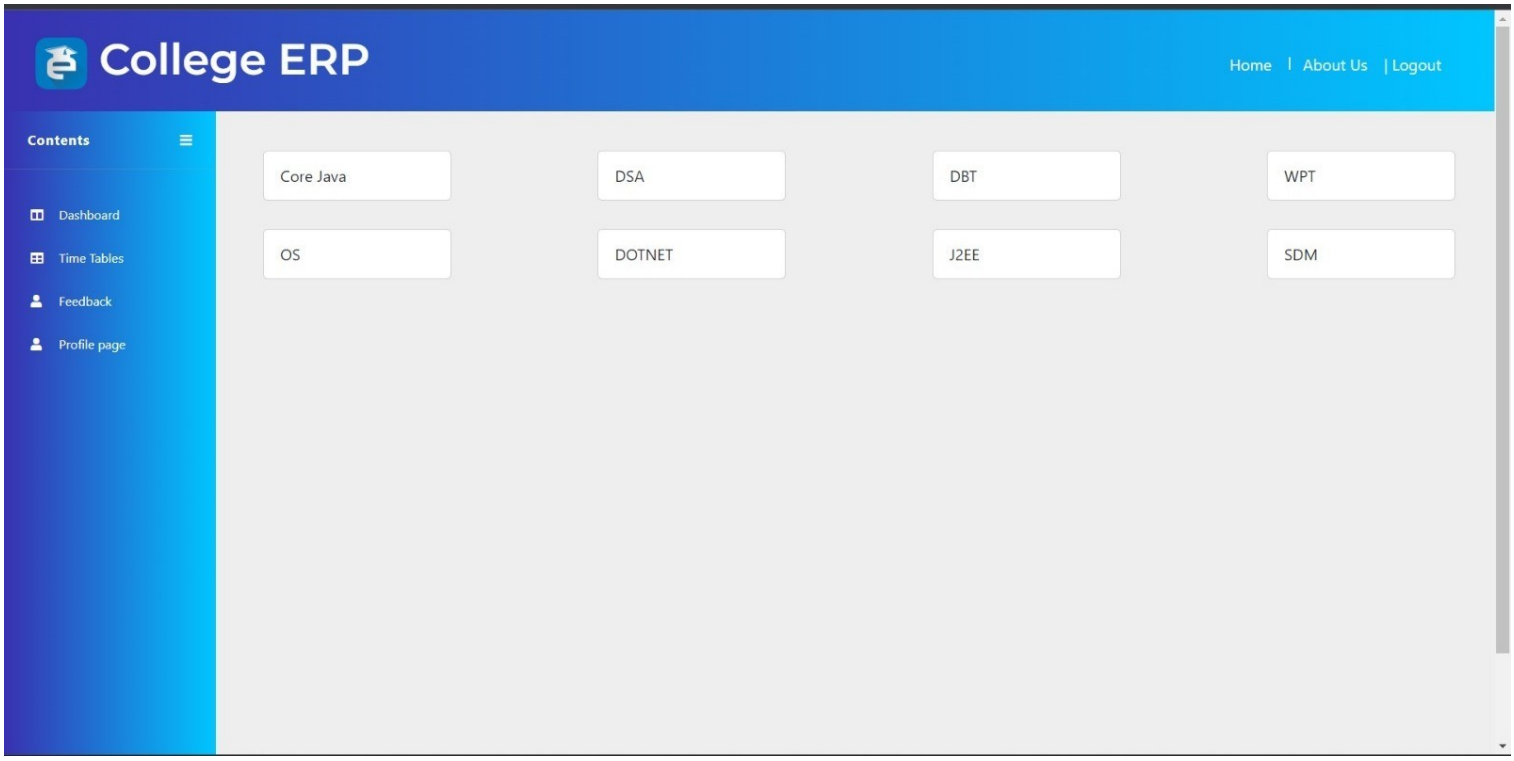


On successful login, student enters into Dashboard

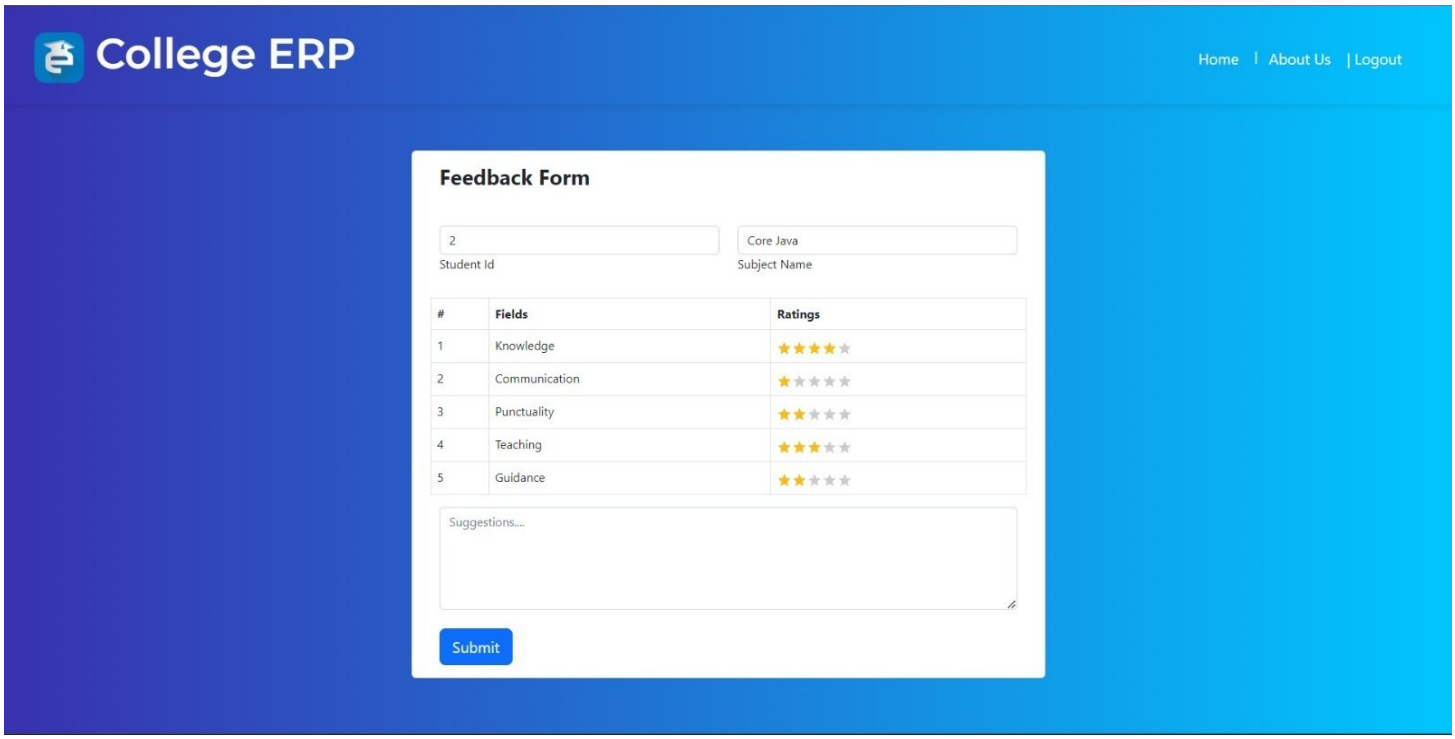

-Here the student can see his/her attendance till date in different modules in percentage.
-Green color shows the percentage of classes attended.
-Red color shows the percentage of classes didn't attend by the student.



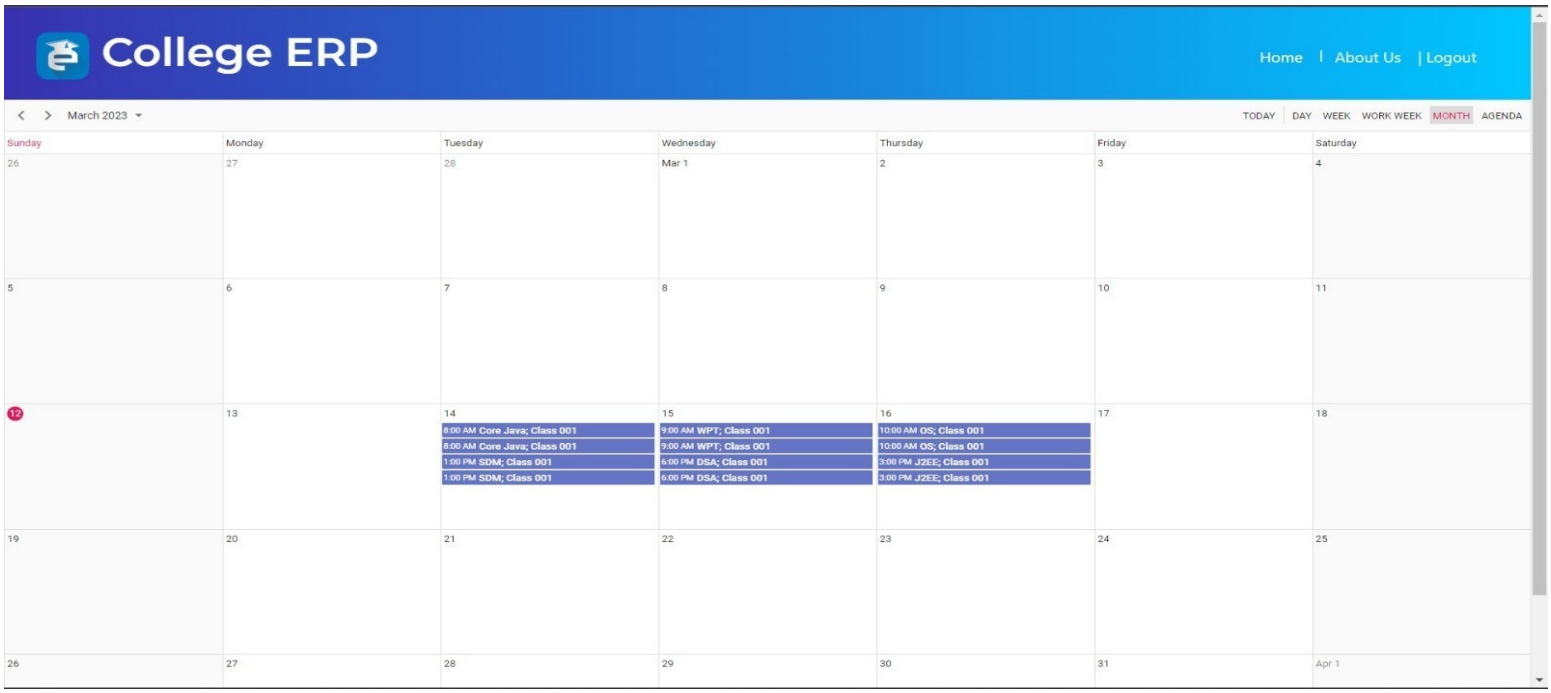

-On navigating to "Profile Page" , Student can view the details he/she has filled during sign up.
-Based on the gender selected Automatic profile animation will appear.



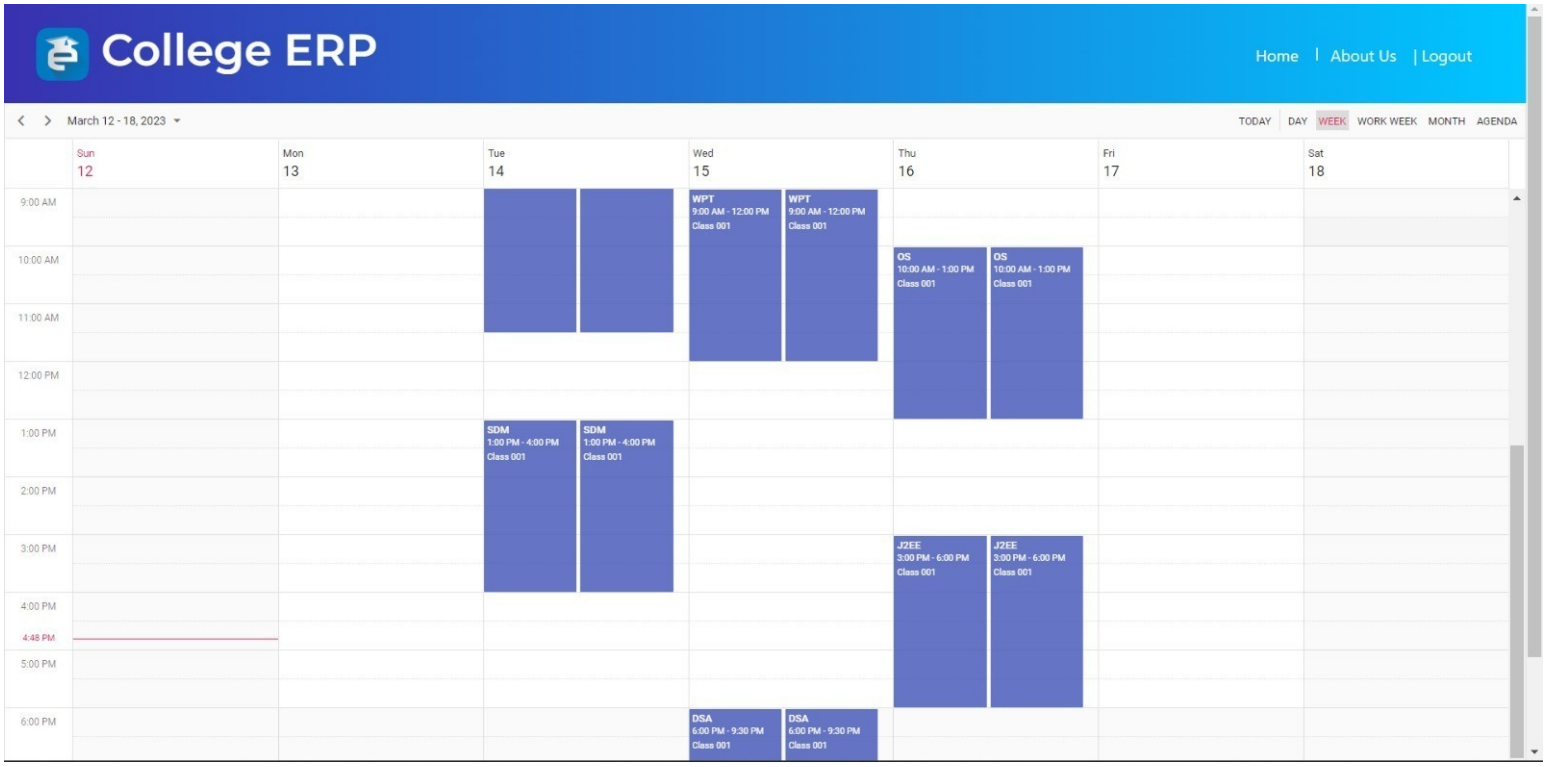

-On navigating to Feedback page.
-Student can select the module for which they want to give feedback.



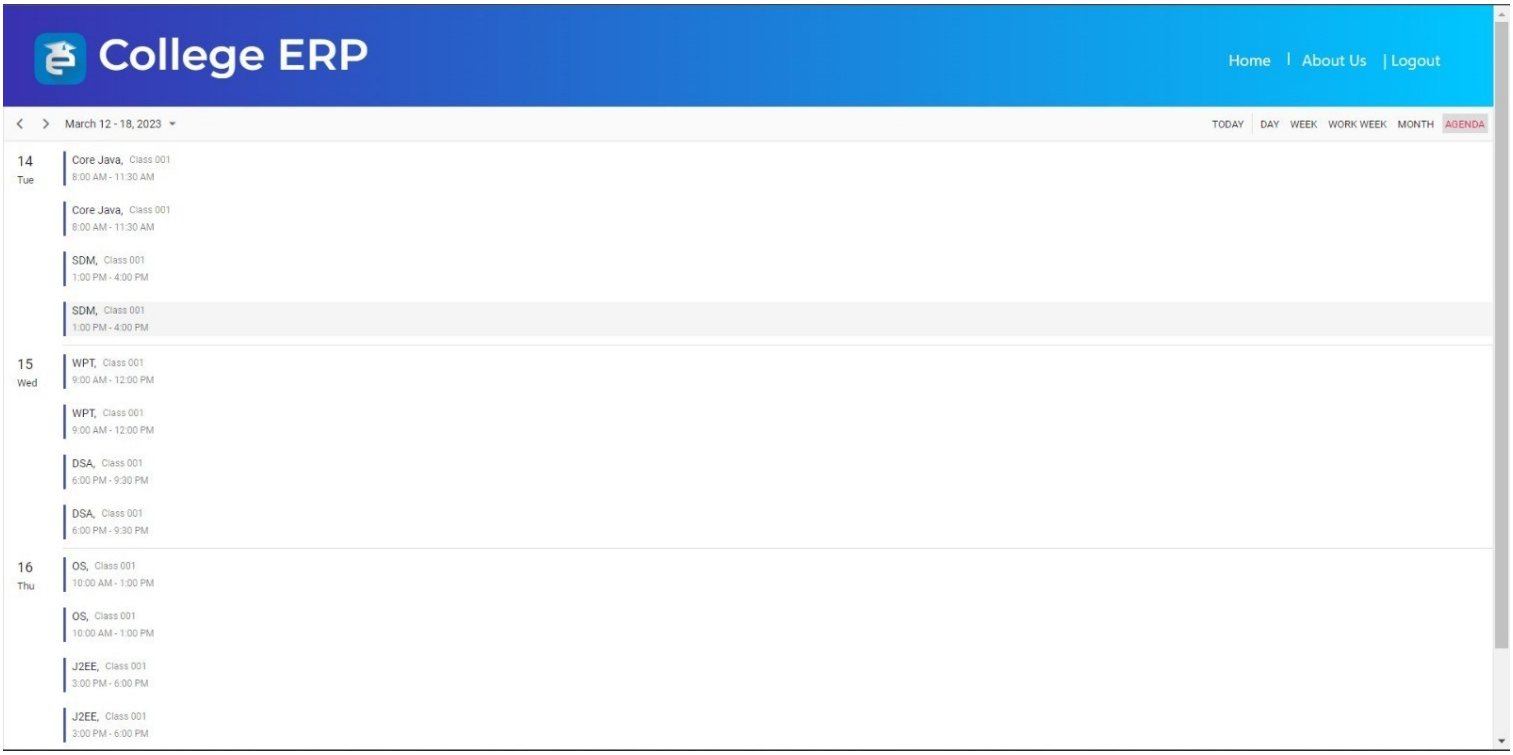
-On navigating to “Time – Table” , student will be able to see the calendar of the entire month and the classes schedule on them.
-Student can change this view to “Week-wise” , “Day-Wise” or even for the entire year.
-By default it will be MONTH wise.



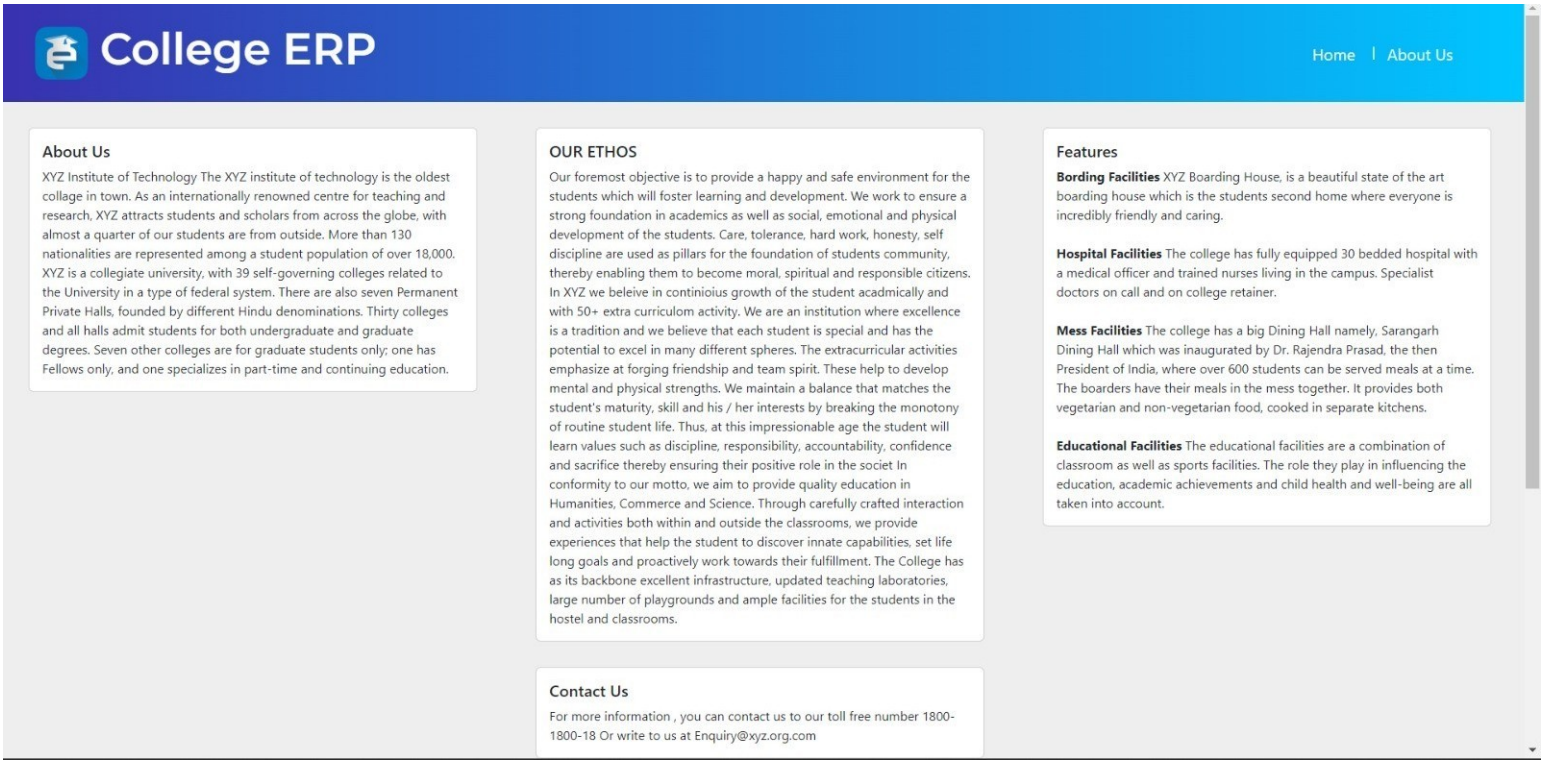

-WEEK wise



-DAY wise


College ERP
Home | About Us | Logout
March 12 - 18, 2023
TODAY DAY WEEK WORK WEEK MONTH AGENDA
14 Tue
Core Java, Class 001
8:00 AM - 11:30 AM
Core Java, Class 001
8:00 AM - 11:30 AM
SDM, Class 001
1:00 PM - 4:00 PM
SDM, Class 001
1:00 PM - 4:00 PM
15 Wed
WPT, Class 001
9:00 AM - 12:00 PM
WPT, Class 001
9:00 AM - 12:00 PM
DSA, Class 001
6:00 PM - 9:30 PM
DSA, Class 001
6:00 PM - 9:30 PM
16 Thu
OS, Class 001
10:00 AM - 1:00 PM
OS, Class 001
10:00 AM - 1:00 PM
J2EE, Class 001
3:00 PM - 6:00 PM
J2EE, Class 001
3:00 PM - 6:00 PM

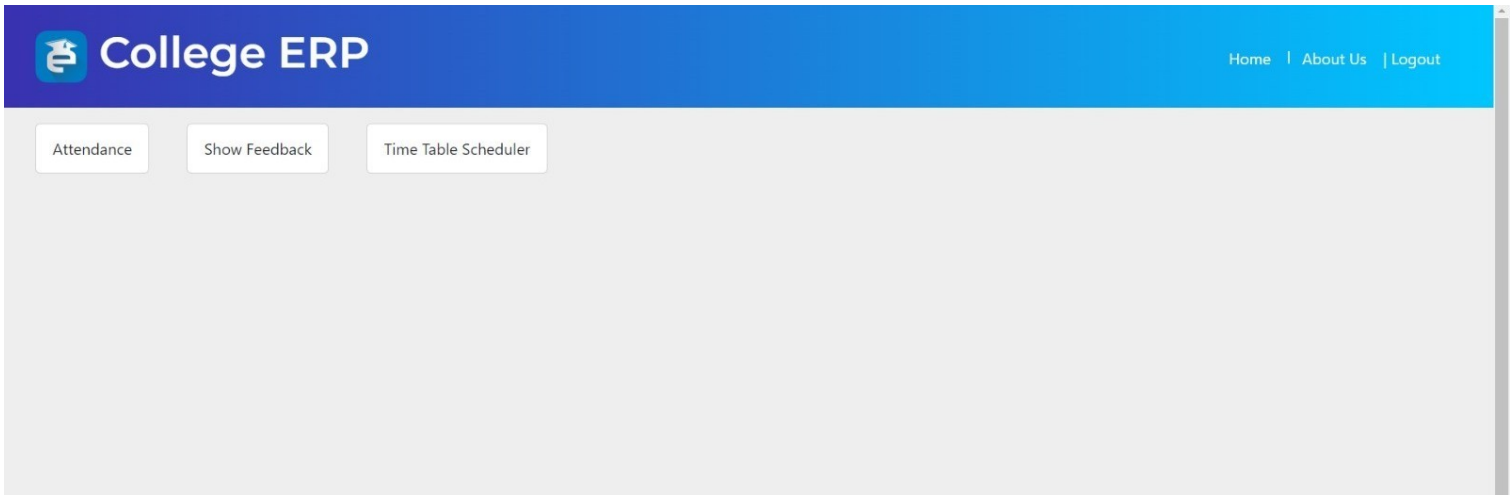
-Apart from these , Student can also navigate to “About” page which contains the information about the Institution.



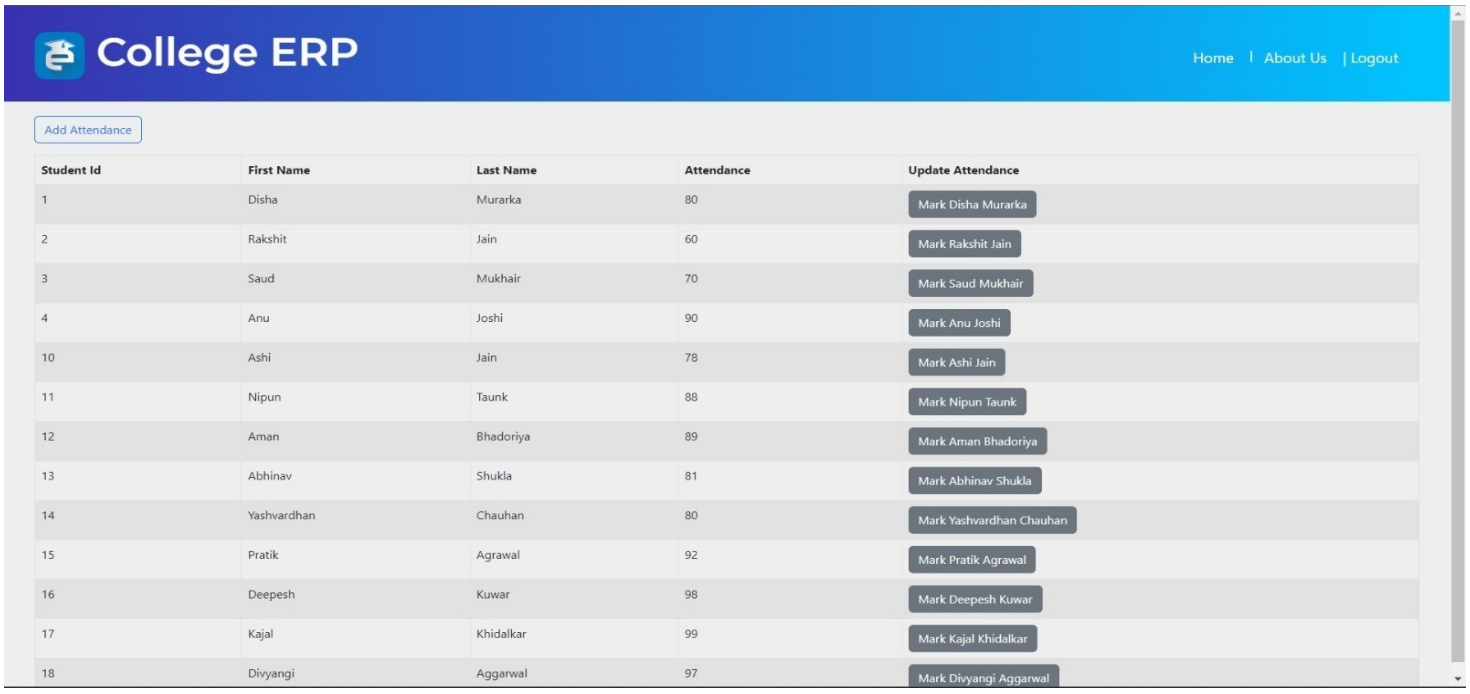
# ADMIN

-When Admin Login, It will take him/her to Admin DashBoard.

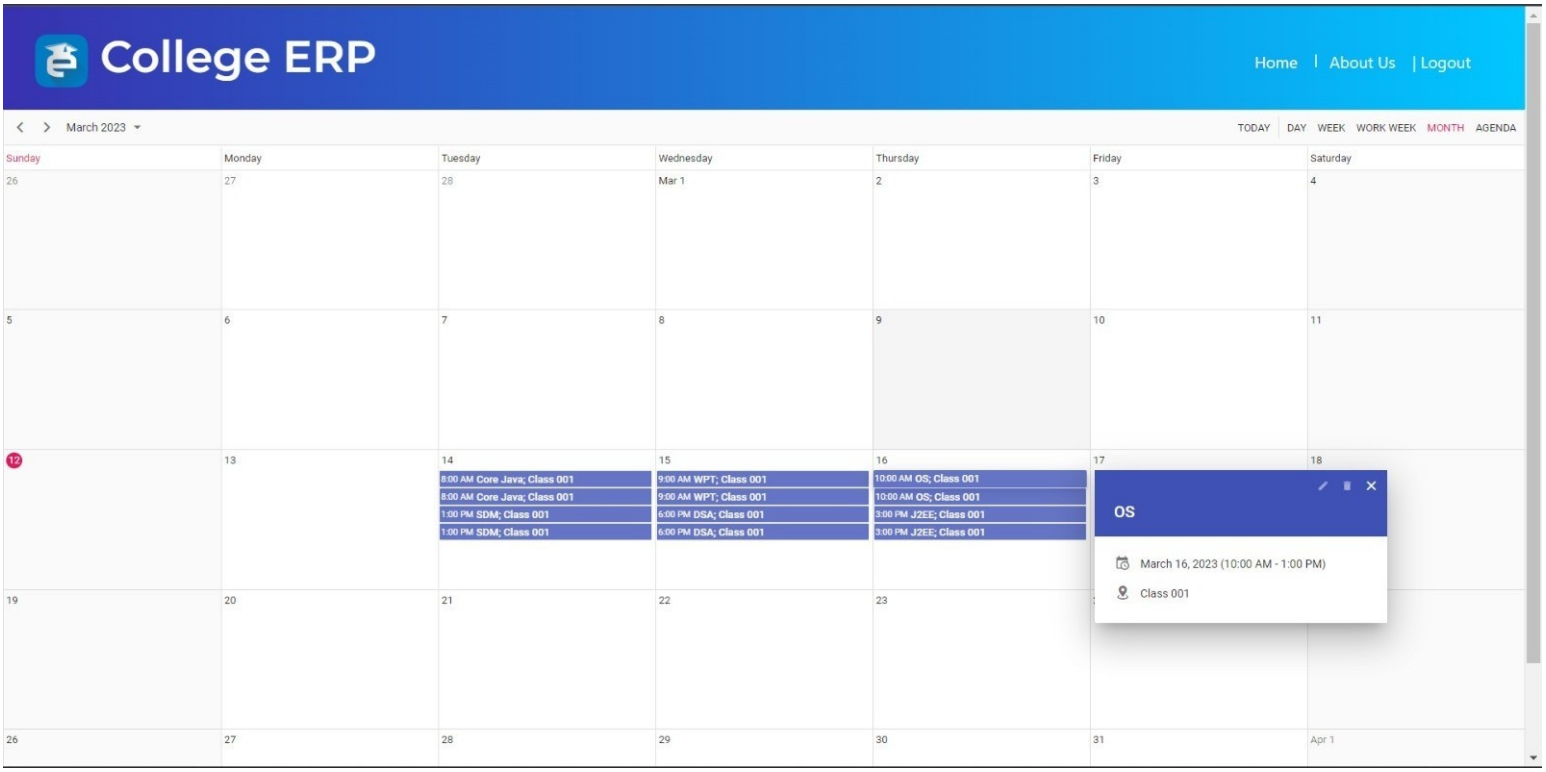
## Admin Dashboard

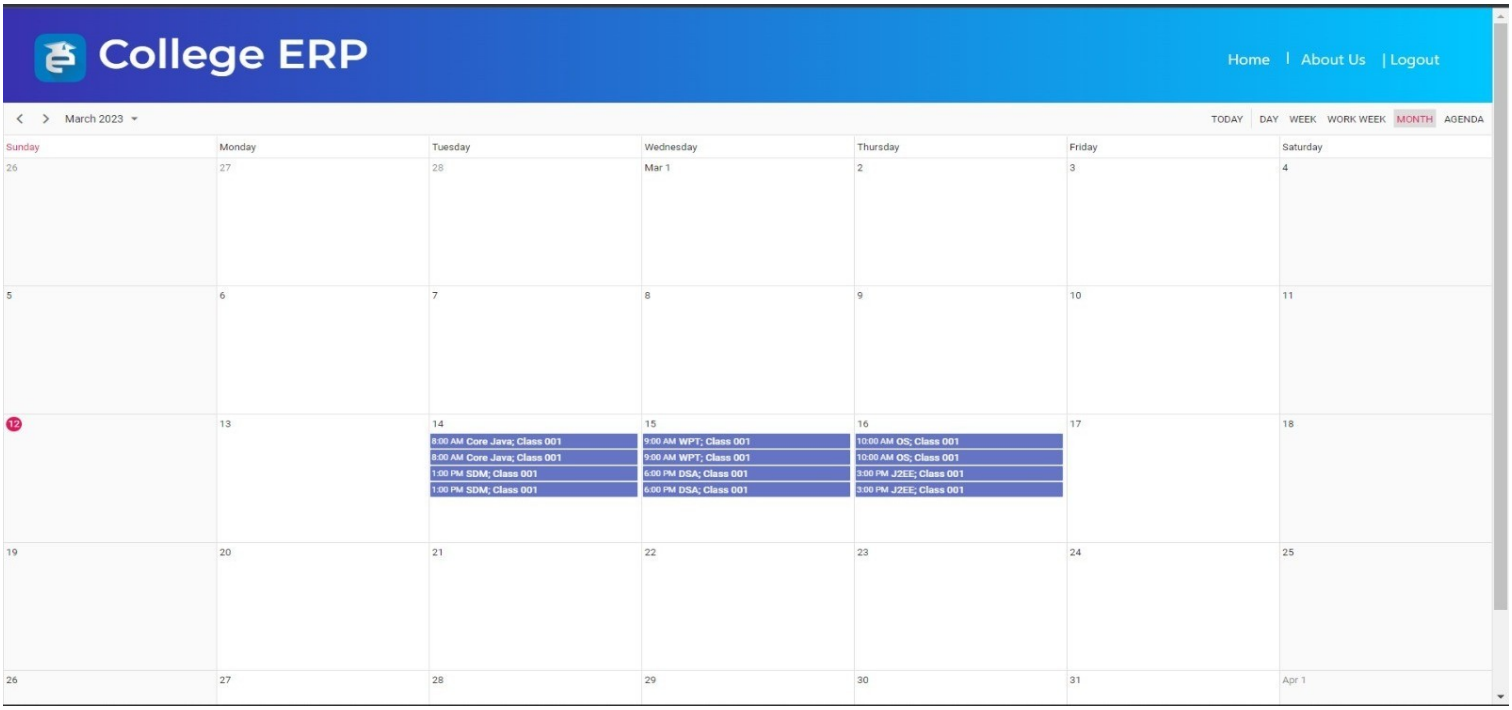


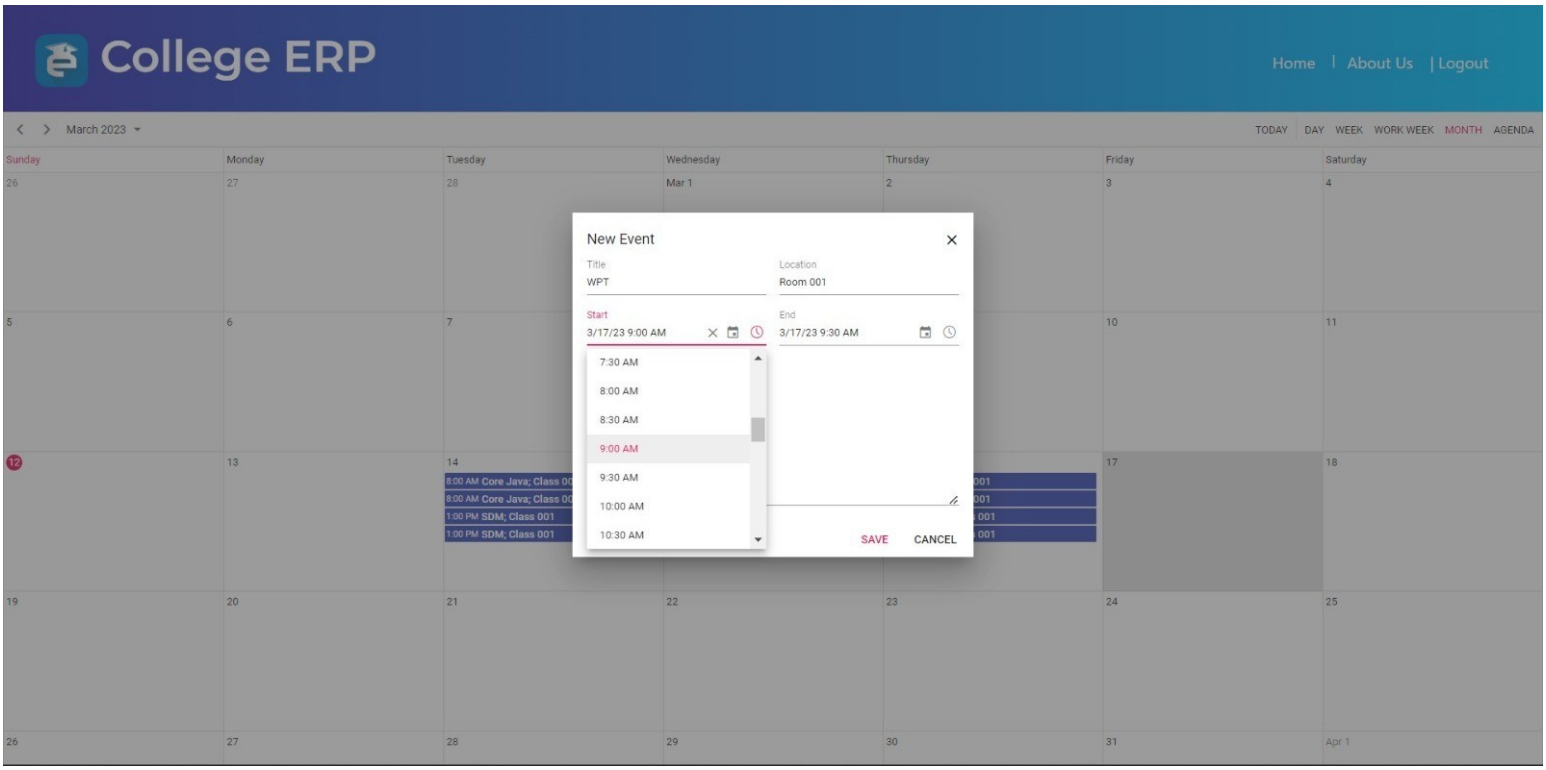

-Admin can add Attendance for each student for particular Subject of different module.

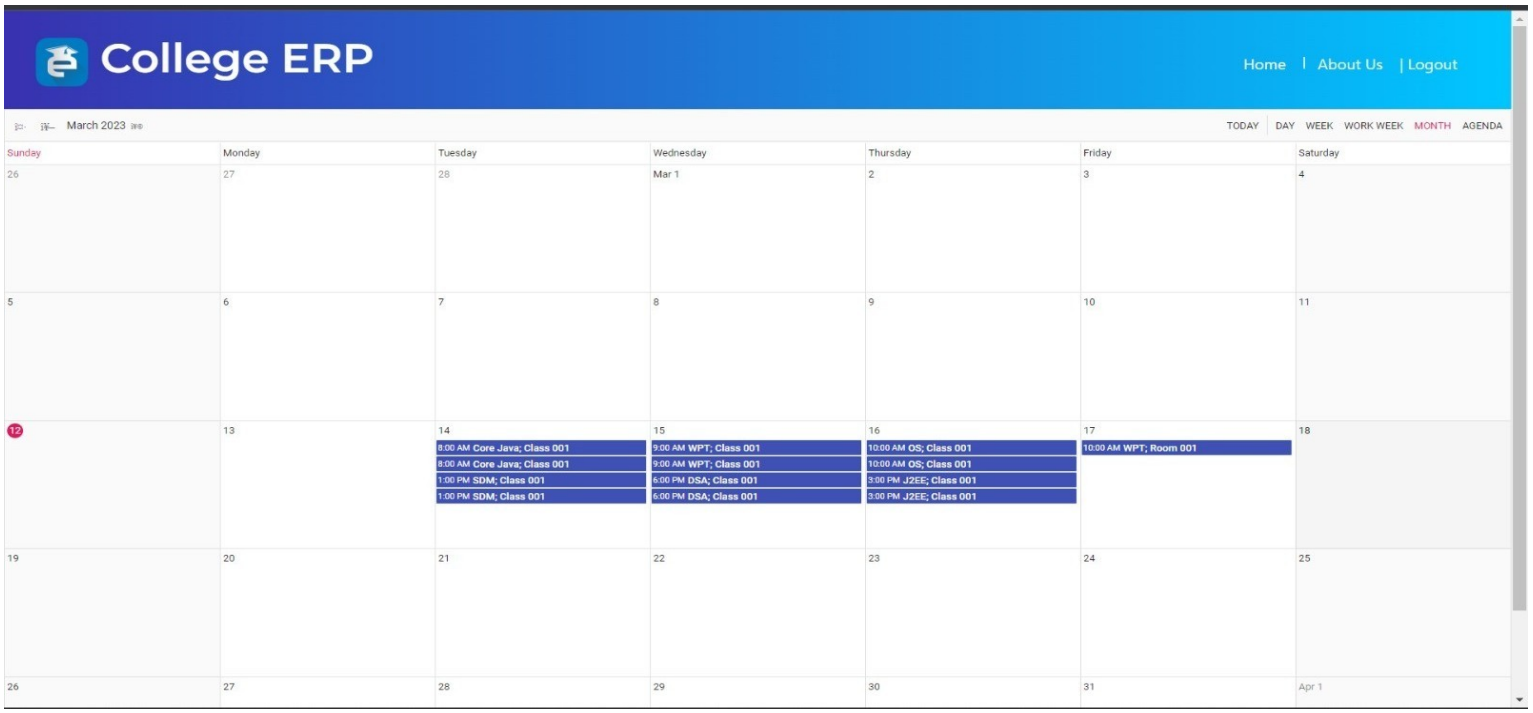


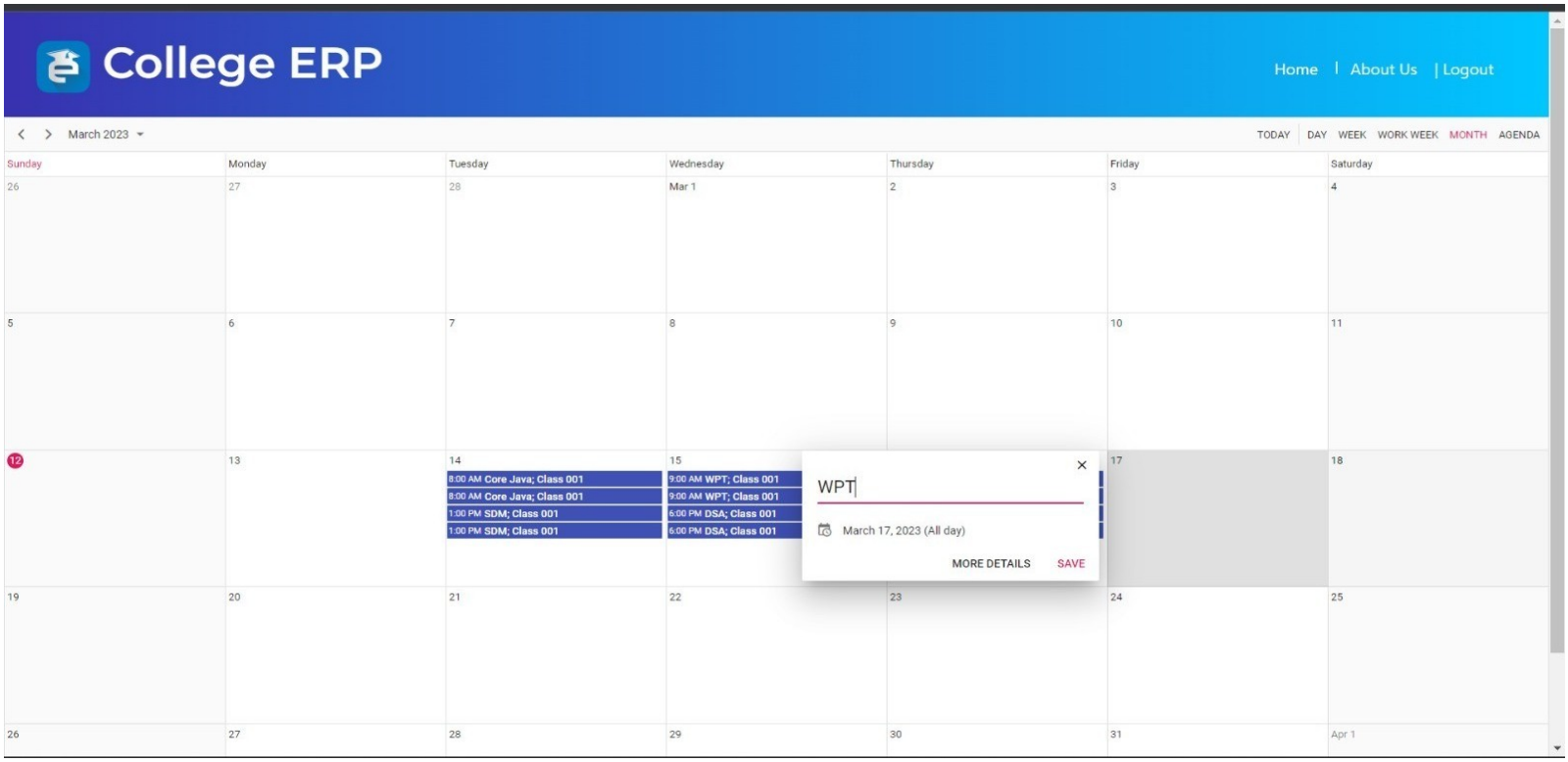
-Admin Can Also Add, Delete , Create and View the Schedules.




College ERP
Home | About Us | Logout
March 2023
TODAY DAY WEEK WORK WEEK MONTH AGENDA
Sunday Monday Tuesday Wednesday Thursday Friday Saturday
New Event
Title
WPT
Location
Room 001
Start
3/17/23 9:00 AM
End
3/17/23 9:30 AM
7:30 AM
8:00 AM
8:30 AM
9:00 AM
9:30 AM
10:00 AM
10:30 AM
SAVE
CANCEL


College ERP
Home | About Us | Logout
March 2023
TODAY DAY WEEK WORK WEEK MONTH AGENDA
Sunday Monday Tuesday Wednesday Thursday Friday Saturday
8:00 AM Core Java; Class 001
1:00 PM SDM; Class 001
9:00 AM WPT; Class 001
6:00 PM DSA; Class 001
10:00 AM OS; Class 001
3:00 PM J2EE; Class 001
10:00 AM WPT; Room 001


College ERP
Home | About Us | Logout
March 2023
TODAY DAY WEEK WORK WEEK MONTH AGENDA
Sunday Monday Tuesday Wednesday Thursday Friday Saturday
26 27 28 Mar 1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 17 18
8:00 AM Core Java; Class 001
8:00 AM Core Java; Class 001
1:00 PM SDM; Class 001
1:00 PM SDM; Class 001
9:00 AM WPT; Class 001
9:00 AM WPT; Class 001
6:00 PM DSA; Class 001
6:00 PM DSA; Class 001
WPT
March 17, 2023 (All day)
MORE DETAILS SAVE
19 20 21 22 23 24 25
26 27 28 29 30 31 Apr 1

# 7. TESTING

One of the main purposes of testing is to validate and verify that the system works as intended. No program or system design is perfect. However, if we implement the system without proper testing, then it may cause problems and lead to a bad user experience.

Testing and checking outcomes of each test gives us the best chance to detect and correct errors before the system is implemented in a production environment.

In the course of our project, we made an effort to manually test each component. In all cases, we obtained the desired results as demonstrated below.

# A. STUDENT FEATURES TEST

| # | Description | Outcome | Result |
|---|---|---|---|
| 1. | Sign up | Student will be giving some personal details and will be able to sign up.<br>Student entry made successfully in database. | Passed |
| 2. | Login as Student | User Id of students are fetched from database, otherwise login failure message will be printed accordingly. | Passed |
| 3. | View Profile | All the Detail entered by student during sign Up viewd. | Passed |
| 4. | Give feedback | Student was Able to give feedback | Passed |
| 5. | View Time-Table | Student was Able to see timetable | Passed |
| 6. | View Attendance | On the dashboard itself it was visible per subject. | Passed |
| 7. | Get Schedule | Current Day Schedule is displayed on the dashboard. | Passed |
| 8. | Log Out | Student was able to log out successfully and came to LOGIN page. | Passed |
| | | | |
| | | | |
| | | | |
| | | | |

# B. ADMIN FEATURES TEST

| | | | |
|---|---|---|---|
| 1. | Login as Admin | Fetched authenticated user details saved in database. | Passed |
| 2. | Add attendance | Attendance added to the database . | Passed |
| 3. | Update attendance | Attendance updates to the Database. | Passed |
| 4. | View Feedback-List | Feedback for different subjects were visible | Passed |
| 5. | Generate Schedule | Admin could successfully added the Event. Updated into the database as well. | Passed |
| 6. | Update Schedule | Admin could update it as well. Changes reflected into database. | Passed |
| 7. | Delete Schedule | Admin could delete well. Changes reflected into database. | Passed |
| 8. | View Schedule | Admin can view the schedule. | Passed |
| 9. | Log Out | Successfully logged out. | Passed |
| | | | |
| | | | |

# 8. CONCLUSION

"CERP", an online student management portal, developed by our project team to provide a platform for students belong to various school, college and educational institutes, to simplify the Admin work. We tried using the latest technologies that are cross-platform and robust. Almost all the software we used was open-source in nature, which keeps the cost of production at a minimum.

We were also meticulous about the user experience aspect of our application so that navigating our website is an easy and seamless experience.

In conclusion, "CERP" as a portal would definitely be beneficial for various colleges, schools or educational institutes for managing their students in Online Mode efficiently. We are confident that the numerous features and visually appealing look of the portal will definitely make this portal effective for all the students and admins. This portal can be scaled on higher side to add more features to it.

# 9. FUTURE SCOPE

Using whatever we have learn over the duration of this course, we tried to make our project as user-friendly and gave it as many features as possible in the limited time allotted for the project work. That said, there are certainly more features that can be added to our application. Some of those are mentioned below:

1. Admin functionalities can be improved for larger scale.
2. More number of courses can be added in the app.
3. If a user forgets password, OTP can be sent to his/her registered mobile number and registered Email Id to reset password.
4. Students can also view their results and will be able to access the course repository wherein they will have all the necessary study material required.
5. Students will be able to submit their assignments and give online tests on the portal itself.
6. We can also add feature like ‘community center’ for all the students of the institution.
7. ALUMNI section can also be introduced on the main page.
8. CAPTCHA can be added to login page.
9. Overall portal can be built on a higher scale so that all functionalities of student management for all courses offered in institution work inside this one portal only.

# 10. REFERENCES

Following is the list of websites we referred during the course of our project:


1. https://getbootstrap.com/docs/5.1/getting-started/introduction/
2. https://www.baeldung.com/
3. https://www.w3schools.com/
4. https://docs.spring.io/spring-data/jpa/docs/current/reference
5. https://javaee.github.io/javaee-spec/javadocs/
6. https://javadoc.io/doc/org.springframework.data/spring-data-jpa/latest/index.html