

Project Name: Online Movie Ticket Booking System

C-DAC March-2022

Documentation On

**“ONLINE MOVIE TICKET BOOKING SYSTEM”**

C-DAC MARCH 2022

**Guided By: Bakul Joshi**

Submitted By :

**Group No: 01**

* **Abdulvahab Maner 220343020003**
* **Sangram Desai 220343020026**
* **Aditya Hande 220343020033**
* **Vyankatesh Nakate 220343020114**

**Table of Contents**

**1. Introduction 4**

1.1 Document Purpose **4**

1.2 Project Background **4**

1.3 Aim & Objectives **4**

**2. Business Requirements Overview**…………………………………………………………**5**

**3.Functional Requirements** **6**

3.1 Admin Module **6**

3.2 Theater Owner Module **6**

3.3 Customer Module **6**

**4.** **Non-Functional Requirement………………………………………………………………7**

**5. Use Case Diagram 8**

5.1 Admin **9**

5.2 Theater Owner **10**

5.3 Customer **11**

**6. Database Design 11**

1 Admin **11**

2 Customer **11**

3 Theaterowner…..……………………………………………………………………… **11**

4 Theater **11**

5 Movies **12**

6 Movies time slots **12**

7 City **12**

8 Login **12**

9 Screen master **13**

10 Screen seat capacity **13**

11 Seat categories …………………………………………………………………..……..**13**

12 Price master ……………………………………………………………………………**13**

13 Screen cat seat allocation……………………………………………………………….**13**

14 Booked Seats…………………………………………………………………………...**14**

15 Time slots……………………………………………………………………………….**14**

**7. E-R Diagram………………………………………………………………………………………14**

**8. Snapshots 15**

**9. Conclusion 20**

**List of Figures**

**Use Case Diagrams 17**

Fig 1 Admin 8

Fig 2 Theater Owner 9

Fig 3 Customer 10

Fig 4 ER Diagram 14

1. **Introduction:**
   1. **Document Purpose:**

This document communicates the business requirements and scope for developing It’s ShowTime e-Movie ticket booking System. The scope of this document is to define the functional and non-functional requirements, business rules, and other constraints requirements.

## Project Background:

It’s ShowTime developing an online movie ticket booking system as a product. On previous days tickets were collected at the counter. Watching movies with family and friends in theatres is one of the best entertainment mediums after a hectic schedule. But all this excitement vanishes after standing in hours in long queues to get tickets booked. Now “It’s ShowTime” enables people to book tickets online 24/7 from anywhere using our It’s ShowTime Application.

The project objective is to book movie tickets on the online portal. The Ticket Booking System is an Internet-based application that can be accessed throughout the internet and can be accessed by anyone who has an internet connection. The user is required to login into the system and needs an online payment system for booking the tickets

## Aim & Objectives:

To provide simple and user-friendly UI to all types of users by Its ShowTime portal. This online ticket booking system provides a website for a cinema hall where any user of the internet can access it. This system also provides an option to cancel previously booked tickets. The website provides complete information regarding currently running movies on all the screens with details on show time and available seats. The online ticket booking system is one of the best opportunities for those who cannot afford enough time to get their tickets reserved while standing in long queues.

1. **Business Requirements Overview:**

* It’s ShowTime application System is a public web application.
* Its’s ShowTime application System will be opened to the State, but in phase 1, the main target is one city.
* There are mainly two types of users i.e. Admin and Customer.
* The users will search for the movie, then for theatre, and then book the tickets online.
* The Admin maintains the user details, movie details, and theatre details, and checks the number of seats available.
* The user should first register and create an account to use the system. Users can log in and can update personal details.
* The user gets all information like Movie details and theatre Details.
* Registered Users can write reviews and give ratings for the movie.
* Any user of the system can read the review about the movie.
* Its ShowTime application System could be maintained by the Admin.

# Functional Requirements Overview:

Its ShowTime System consists of two modules described below.

1. Admin Module
2. Customer Module
3. Theatre owner Module

# **3.1 Admin** **Module**

* + Login: Admin logins to the system by entering a valid user id and password to manage the system.
  + Verification: Admin can verify theatre owner.
  + Remove theatres owners: Admin can remove the theatre owners.
  + Add movies: The system shall have a feature for the admin to add movies and their details
  + Logout: After managing the system admin will log out.

# **Theatre** **owner** **Module**

* + Registration: Theatre owner will register on the site.
  + Login: Theatre owners logins into the system by entering a valid user id and password for doing the task.
  + Logout: After doing his/her task he/she will log out.
  + Add Theater: The owner can add many theaters to his account.
  + Allot Movie to Screen: The owner can allot movies for the theater at a particular screen.
  + Remove movies from the screen: The owner can remove movies from the theater screen.

# **3.3 Customer** **Module**

* Registration: If a customer wants to book the ticket then he/she must be registered, an unregistered user can’t book the ticket.
* Login: Customer logins to the system by entering a valid user id and password for booking the ticket.
* Search Movie: The system shall have a search function. Customers can search for movies based on movie name, date, and time.
* Search theatre: Customers can search for multiple theatres
* Seat Viewing: The customer shall be shown an image of the seats from which the desired seats are selected.
* Ticket Booking: The customer shall be given an option to book a ticket up to one hour before the start of the movie time
* Give feedback: Customers can give feedback to the system.
* Payment: By online mode.
* Logout.

1. **Non-Functional Requirement:**

* **Security**

The system’s back-end servers shall only be accessible to authenticated administrators. Sensitive data will be encrypted before being sent over insecure connections like the internet.

* **Reliability**

The reliability of the overall program depends on the reliability of the separate components. The main pillar of the reliability of the system is the backup of the database which is continuously maintained and updated to reflect the most recent changes. Thus, the overall stability of the system depends on the stability of the container and its underlying operating system.

* **Availability**

The system should be available at all times, meaning the user can access it using a web browser, only restricted by the downtime of the server on which the system runs. A replacement page will be shown in case of a hardware failure or database corruption. Also, in case of a hardware failure or database corruption, backups of the database should be retrieved from the server and saved by the administrator. Then the service will be restarted. It means 24/7 availability.

* **Maintainability**

A commercial database is used for maintaining the database and the application server takes care of the site. In case of failure, a re-initialization of the program will be done. Also, the software design is being done with modularity in mind so that maintainability can be done efficiently.

* **Portability**

The application is HTML and scripting language based. So, the end-user part is fully portable and any system using any web browser should be able to use the features of the system, including any hardware platform that is available or will be available in the future. An end-user is using this system on any OS; either it is Windows or Linux. The system shall run on PC, Laptops.

* **Accessibility**

The system will be a web-based application it is going to be accessible on the web browser.

* **Back up**

We will take a backup in our system database. To enable the administrator and the user to access the data from our system.

* **Performance**

The product shall be based on the web and has to be run from a web server. The product shall take initial load time depending on internet connection strength which also depends on the media from which the product is run. The performance shall depend upon the hardware components of the client/customer.

* **Supportability**

The source code developed for this system shall be maintained in the configuration management tool.

## 5. Use-Case Diagram

**5.1 Admin:**

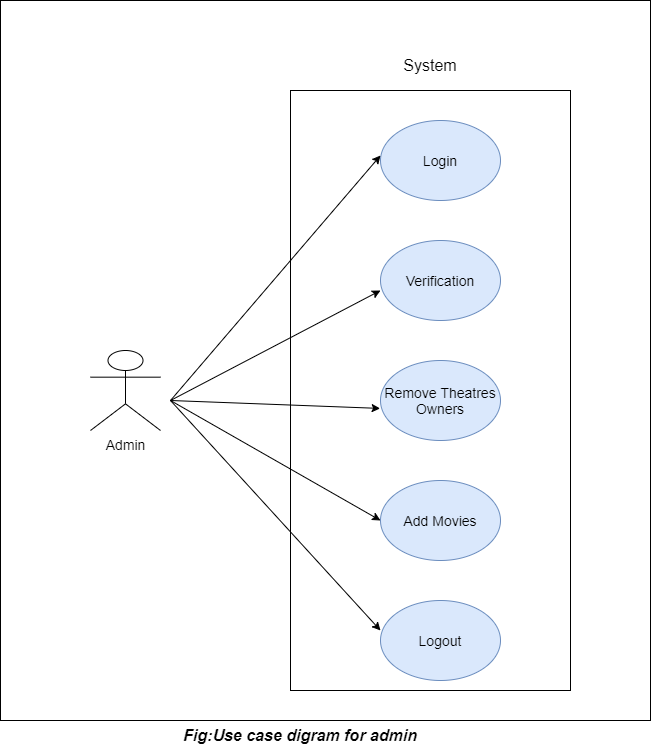
****

Fig. Use-Case Diagram for Admin

## 5.2 Student:

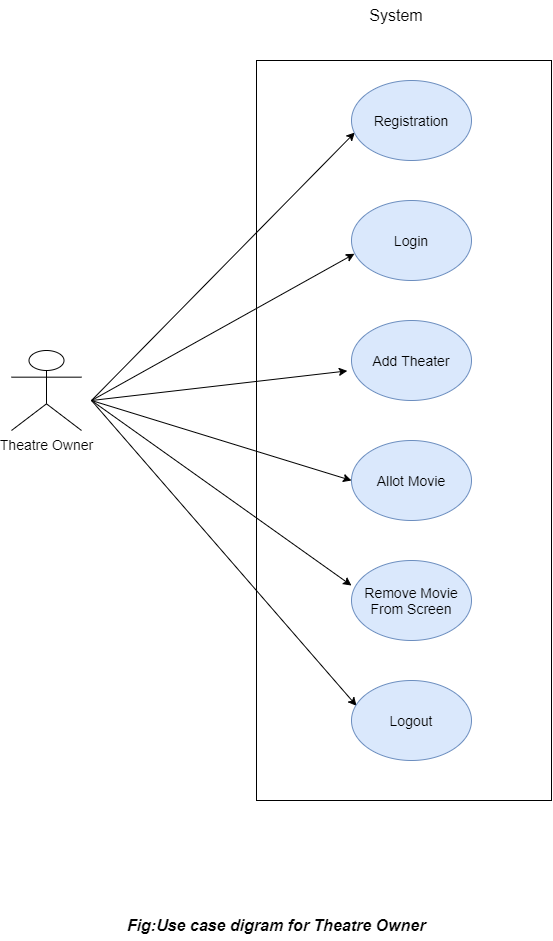
****

Fig. Use-Case Diagram for Theater Owner

**5.3 Customer:**

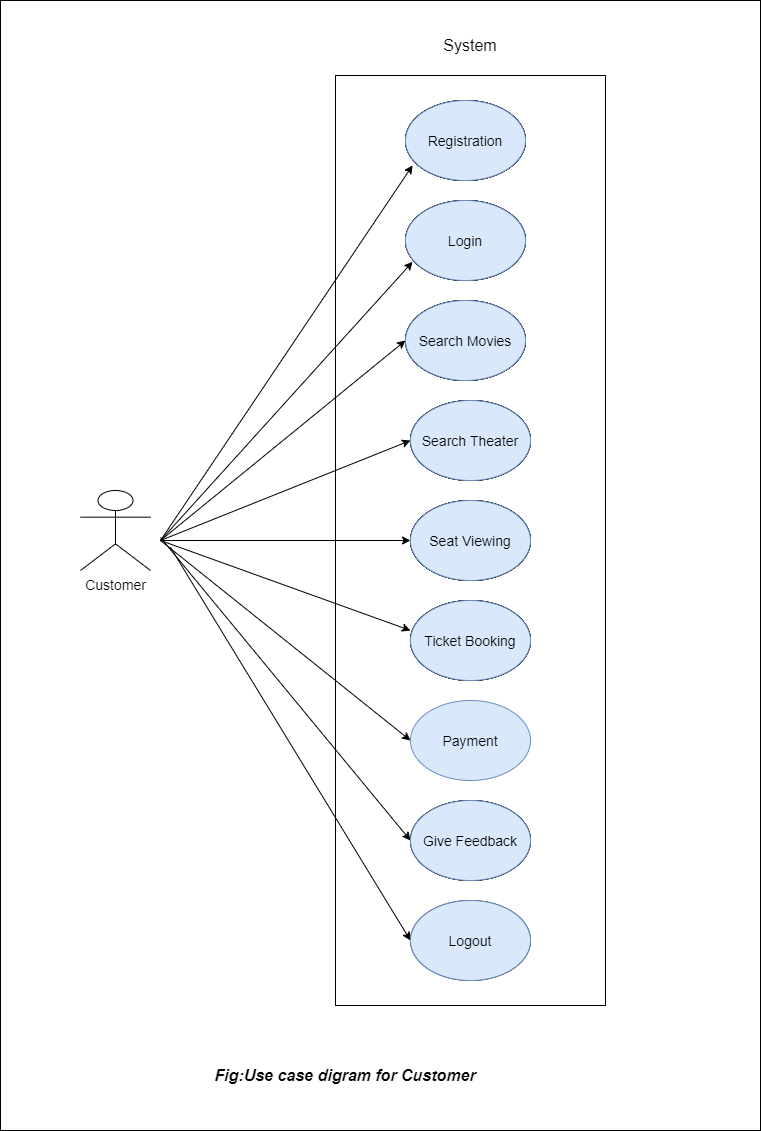


Fig. Use-Case Diagram for Customer

**6. Database Design:**

**1] Admin table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| aid | Integer | No | PK | NULL | admin ID |
| uid | Integer | No | FK | NULL | User ID |
| name | varchar(30) | No |  | NULL | Full Name |
| address | varchar(100) | No |  | NULL | Address |

**2] Customer table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| cid | Integer | No | PK | AI | Customer ID |
| uid | Integer | No | FK |  | User ID |
| cname | Varchar(45) | No |  |  | Full Name |
| caddress | Varchar(100) | No |  |  | Address of Customer |
| birthdate | date | No |  |  | Birthdate of Customer |
| gender | Varchar(10) | No |  |  | Gender of customer |
| ccontact | Varchar(20) | No |  |  | Contact of customer |
| cemail | Varchar(30) | No |  |  | Email of customer |

**3] theater\_owner table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| oid | Integer | No | PK | AI | Theater owner ID |
| uid | Integer | No | FK |  | User ID |
| name | Varchar(35) | No |  |  | Full Name |
| address | Varchar(45) | No |  |  | Address of theater owner |
| gender | Varchar(6) | No |  |  |  |
| contact | Varchar(15) | No |  |  |  |
| mail | Varchar(45) | No |  |  |  |
| id\_proof | Varchar(45) | No |  |  |  |

**4] theater table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| tid | Integer | No | PK |  | Theater ID |
| oid | Integer | No | FK |  | Theater owner ID |
| cityid | Integer | No | FK |  | Theater city |
| tname | Varchar(35) | No |  |  | Theater Name |
| licence | Varchar(35) | No |  |  | Licence No |
| screens | Integer | No |  |  | No of screens |

**5] movies table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| mid | Integer | No | PK | AI | Movie ID |
| mname | Varchar(45) | No |  |  | Movie Name |
| minfo | Varchar(255) | No |  |  | Movie information |
| cast | Varchar(255) | No |  |  | Movie cast |
| type | Varchar(45) | No |  |  | Movie Type (2D,3D) |
| release\_date | Date | No |  |  |  |
| start\_date | Date | No |  |  |  |
| end\_date | Date | No |  |  |  |
| language | Varchar(45) | No |  |  |  |
| poster | Longblob | Yes |  |  |  |
| tax\_free | Tinyint | No |  |  |  |

**6] movie\_time\_slots table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| dummy\_mts\_id | Integer | No | PK | AI | To make unique record |
| mid | Integer | No | FK |  | Movie ID |
| scnid | Integer | No | FK |  | Screen ID |
| slot\_id | Integer | No | FK |  | Movie slot ID |

**7] city table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| cityid | Integer | No | PK | AI | City ID |
| city\_name | Varchar(45) | No |  |  | City Name |
| city\_pincode | Varchar(45) | No |  |  | City Pincode |

**8] login\_table table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| uid | Integer | No | PK | AI | Unique ID for users |
| userid | Varchar(45) | No | UNI |  | User ID |
| pwd | Varchar(45) | No |  |  | Password of user |
| role | Varchar(45) | No |  |  | Role of users |
| active\_status | Tinyint | No |  |  |  |

**9] screen\_master table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| scnid | Integer | No | PK | AI | Screen ID |
| tid | Integer | No | FK |  | Theater ID |
| screen\_no | Integer | No |  |  | Screen No of theater |

**10] screen\_seat\_capacity table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| dummy\_ssc\_id | Integer | No | PK | AI | To make unique record |
| scnid | Integer | No | FK |  | Screen ID |
| total | Integer | No |  |  | Capacity of theater seats |
| rows\_seats | Integer | No |  |  | No of rows in one screen |
| columns\_seats | Integer | No |  |  | No of column in one screen |

**11] seat\_categories table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| cat\_id | Integer | No | PK | AI | Category ID |
| category\_name | Varchar(45) | No |  |  | Category Name |

**12] price\_master table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| price\_id | Integer | No | PK | AI | Price ID |
| tid | Integer | No | FK |  | Theater ID |
| cat\_id | Integer | No | FK |  | Category ID |
| price | Integer | No |  |  |  |

**13] screen\_cat\_seat\_allocation table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| dummy\_scsa\_id | Integer | No | PK | AI | To make unique record |
| scnid | Integer | No | FK |  | Screen ID |
| cat\_id | Integer | No | FK |  | Category ID |
| seats | Integer | No |  |  |  |

**14] booked\_seats table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| dummy\_bs\_id | Integer | No | PK | AI | To make unique record |
| bid | Integer | No | MUL |  | Booking ID |
| seatid | Varchar(10) | No |  |  |  |

**15] time\_slots table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| slot\_id | Integer | No | PK | AI | Slot ID |
| start\_time | Varchar(20) | No |  |  | Show start time |
| end\_time | Varchar(20) | No |  |  | Show end time |

**16] booking\_master table**

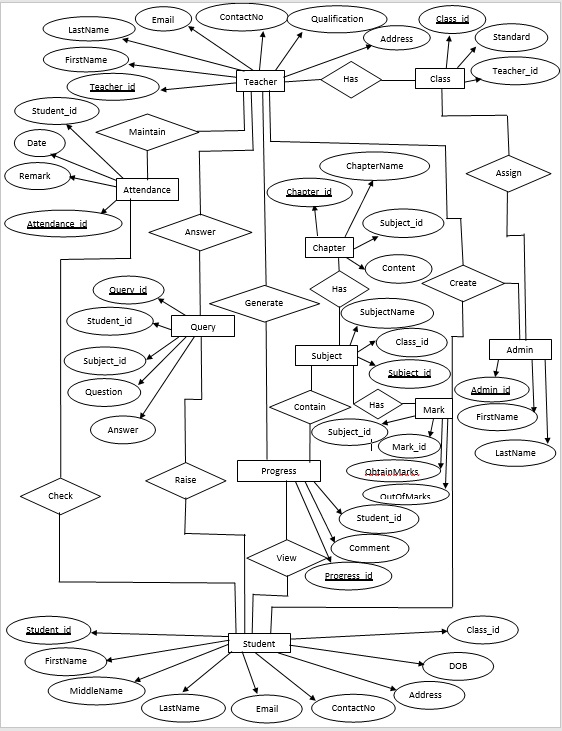
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| bid | Integer | No | PK | AI | Boking ID |
| cid | Integer | No | FK |  | Customer ID |
| mid | Integer | No | FK |  | Movie ID |
| scnid | Integer | No | FK |  | Screen ID |
| slot\_id | Integer | No | FK |  | Movie slot ID |
| booking\_date | Date | No |  |  | Booking date |
| show\_date | Date | No |  |  | Movie show date |
| booked\_seats | Integer | No |  |  | Booked seat no |
| charges | Integer | No |  |  | Ticket Charges |
| payment\_status | Tinyint | No |  |  |  |
| payment\_mode | Varchar(15) | No |  |  |  |
| booked\_status | Tinyint | No |  |  |  |
|  |  |  |  |  |  |

**17] booking\_master table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Description** |
| fid | Integer | No | PK | AI | Feedback ID |
| cid | Integer | No | FK |  | Customer ID |
| rating | Integer | No |  |  | Platform Rating |
| Varchar(15) | Varchar(500) | No |  |  |  |

\

## 7. ER-Diagram:



**E-R diagram shows database of Paperless School Management System**

**8. Snapshots:**

**8.1 Home Page:**

Following snapshot shows the Home page for Paperless School Management System

****

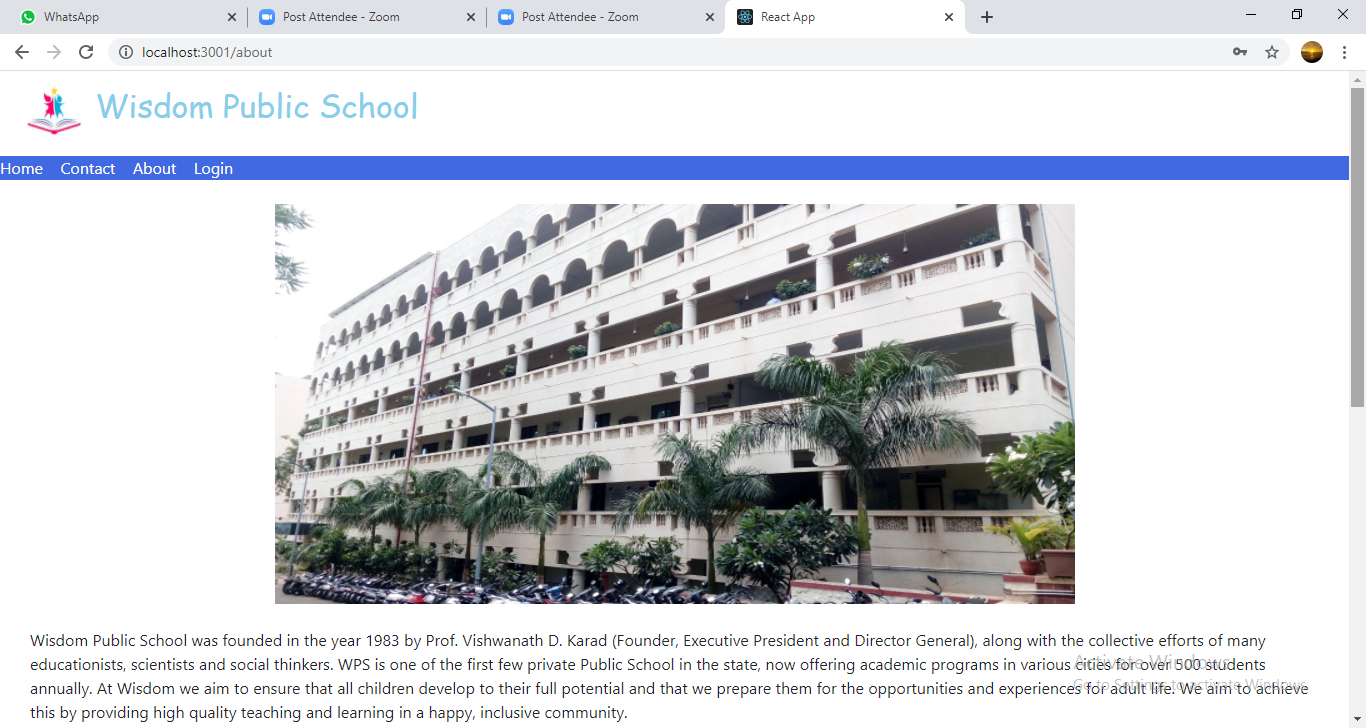
This page contains following controls

* Home
* Contact
* About
* Login Button

**8.2 Contact Page**

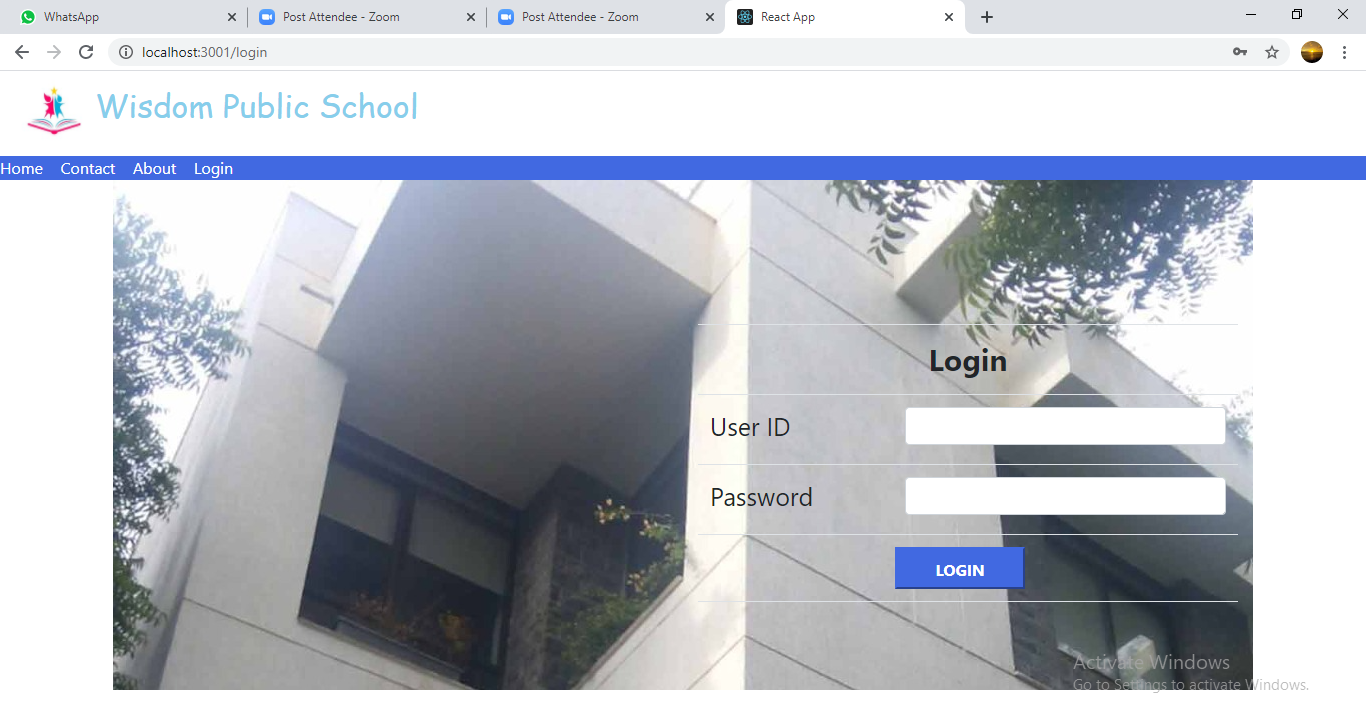


**8.3 About Page**



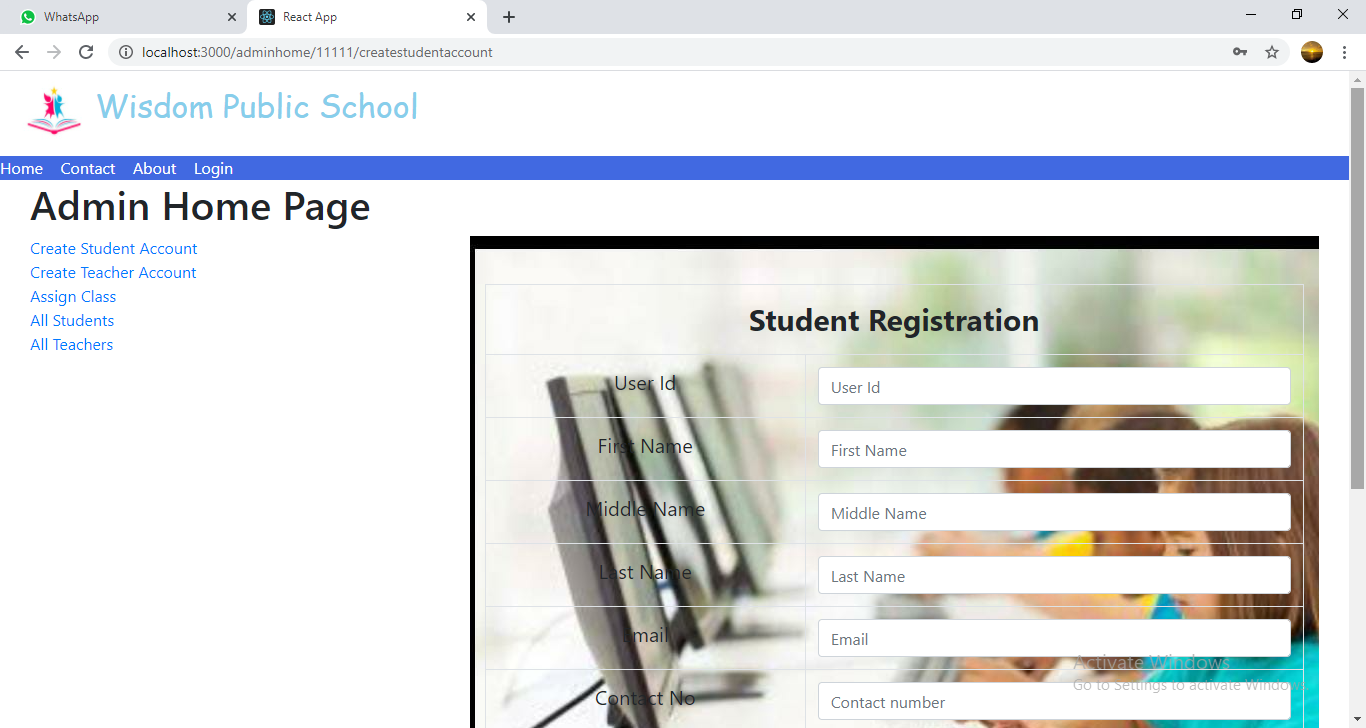
**8.4 Login Page**

Following snapshot shows the Login page for Paperless School Management System

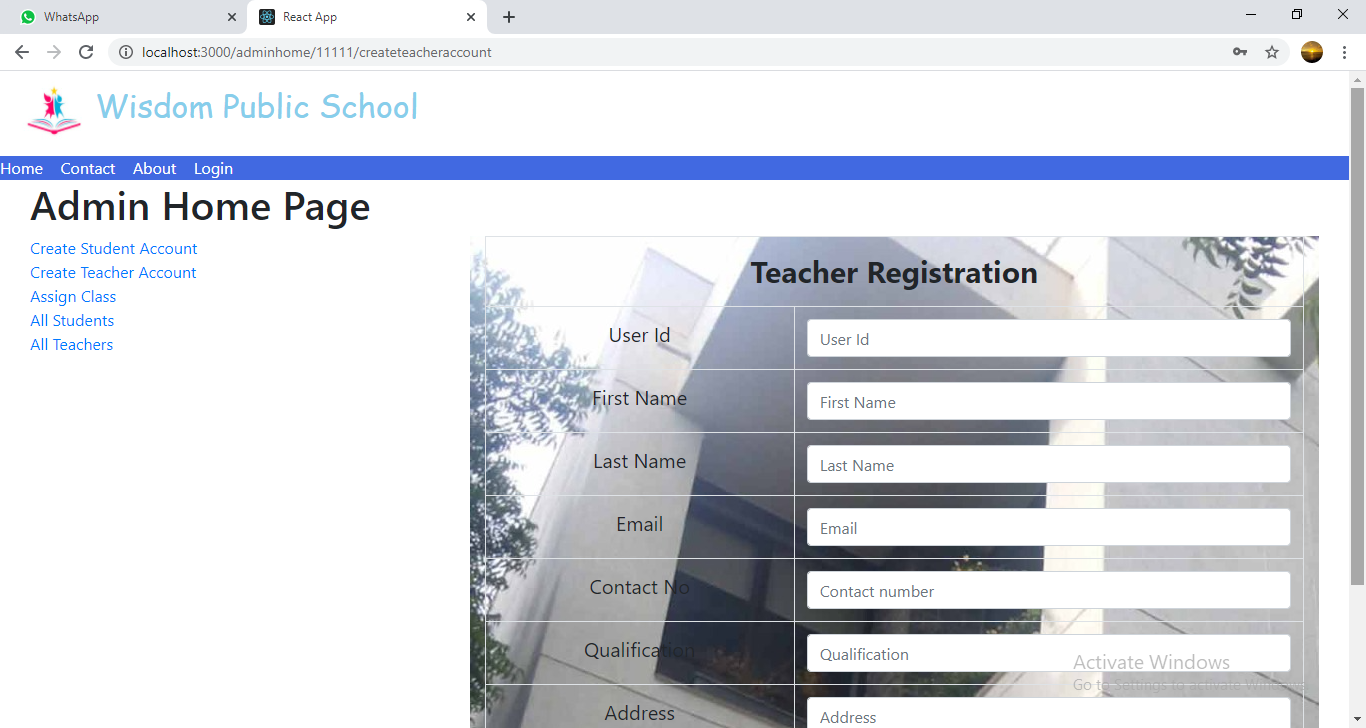


**8.5 Admin Home Page**

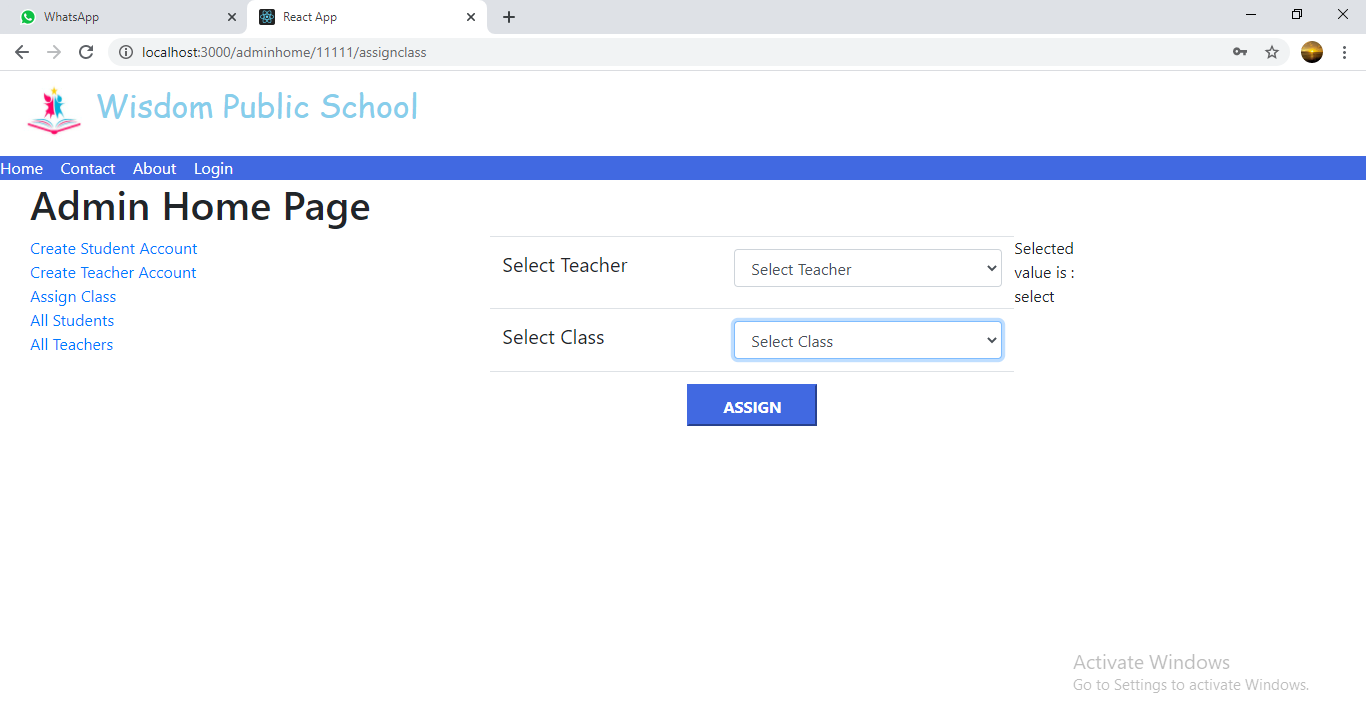
**8.5.1 Student Registration Page**



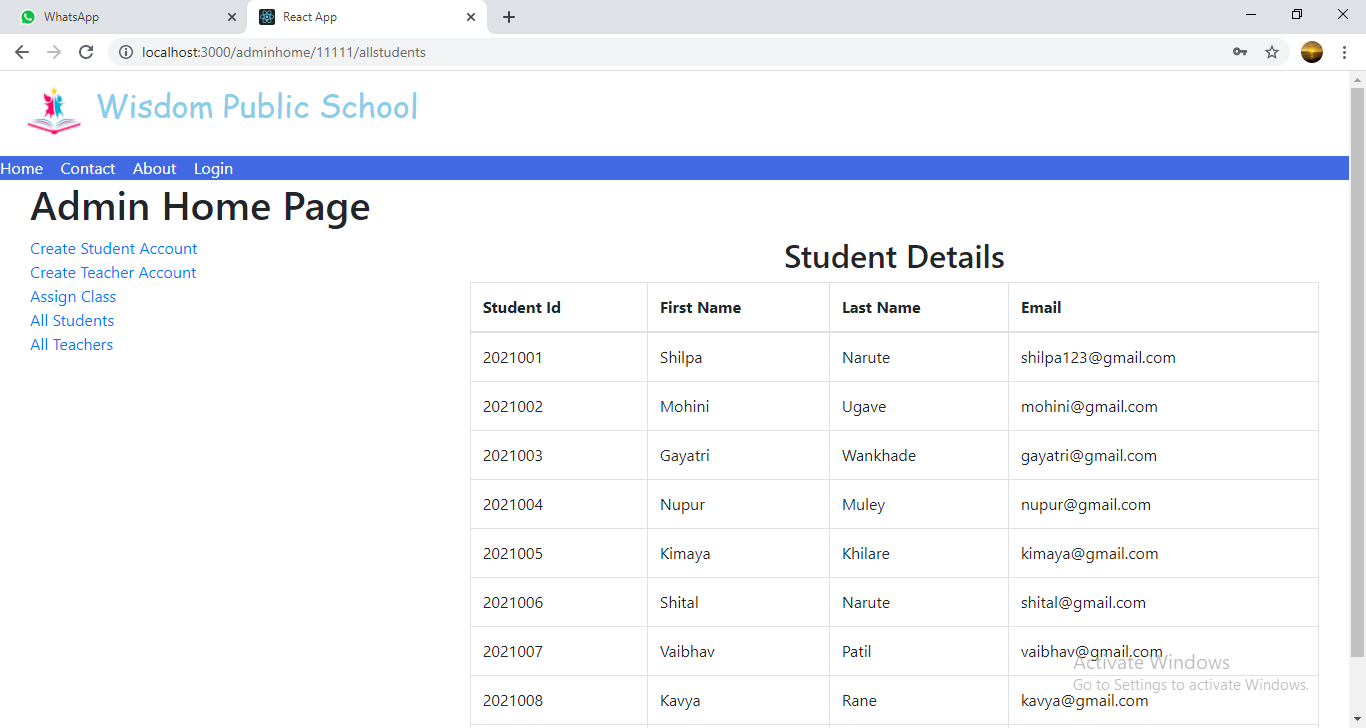
**8.5.2 Teacher Registration Page**



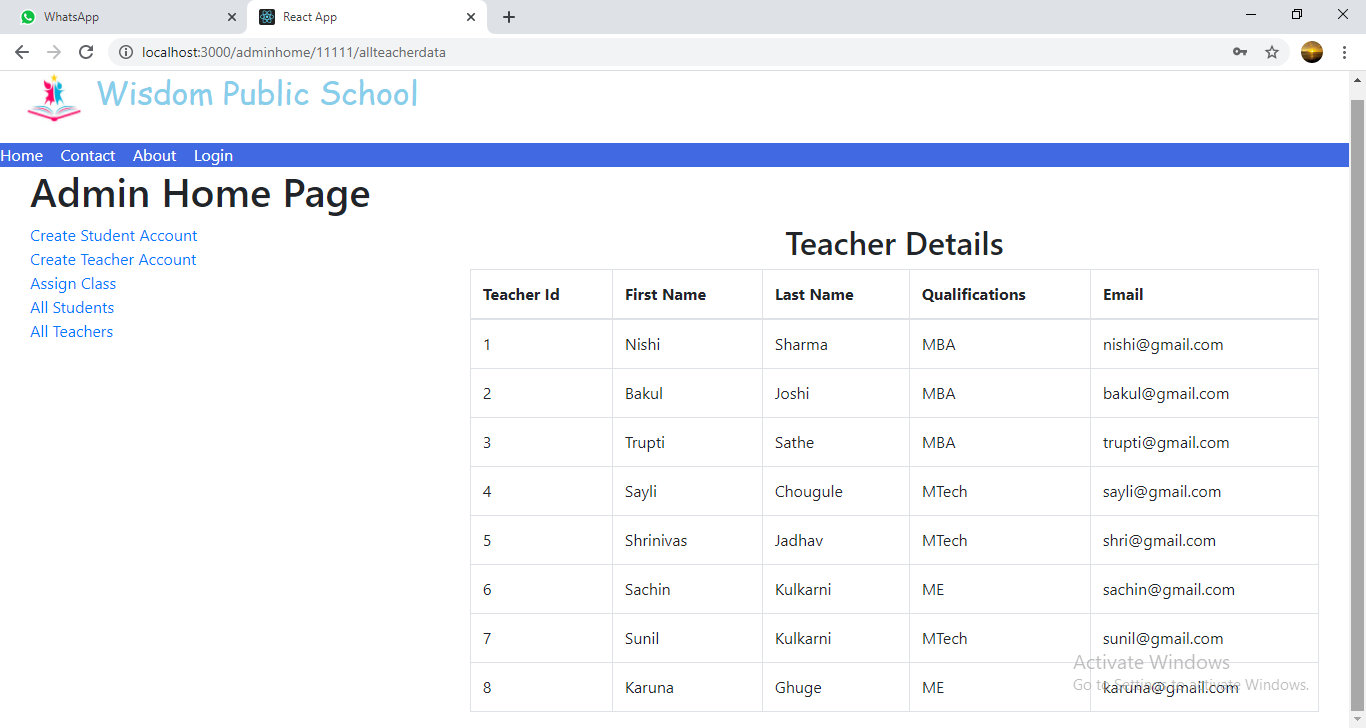
**8.5.3 Assign Class Page**



**8.5.4 All Students List**

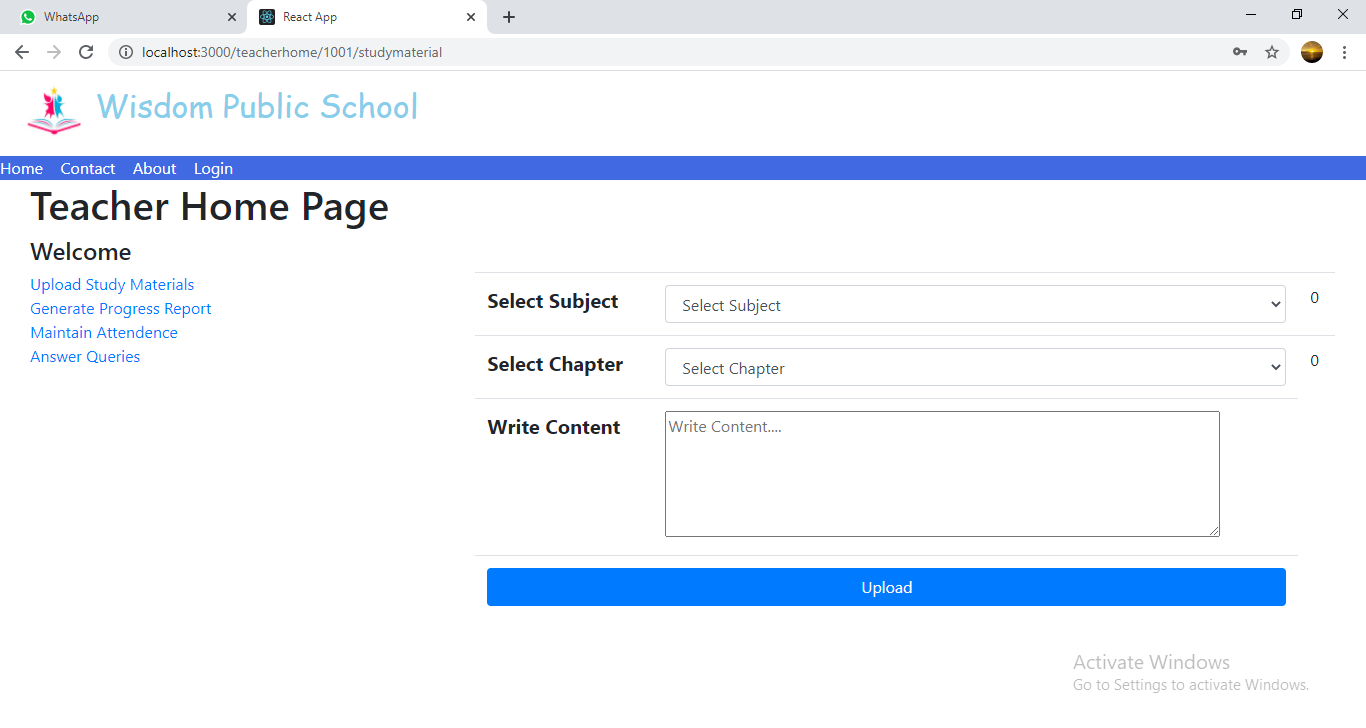


**8.5.5 All Teachers List**

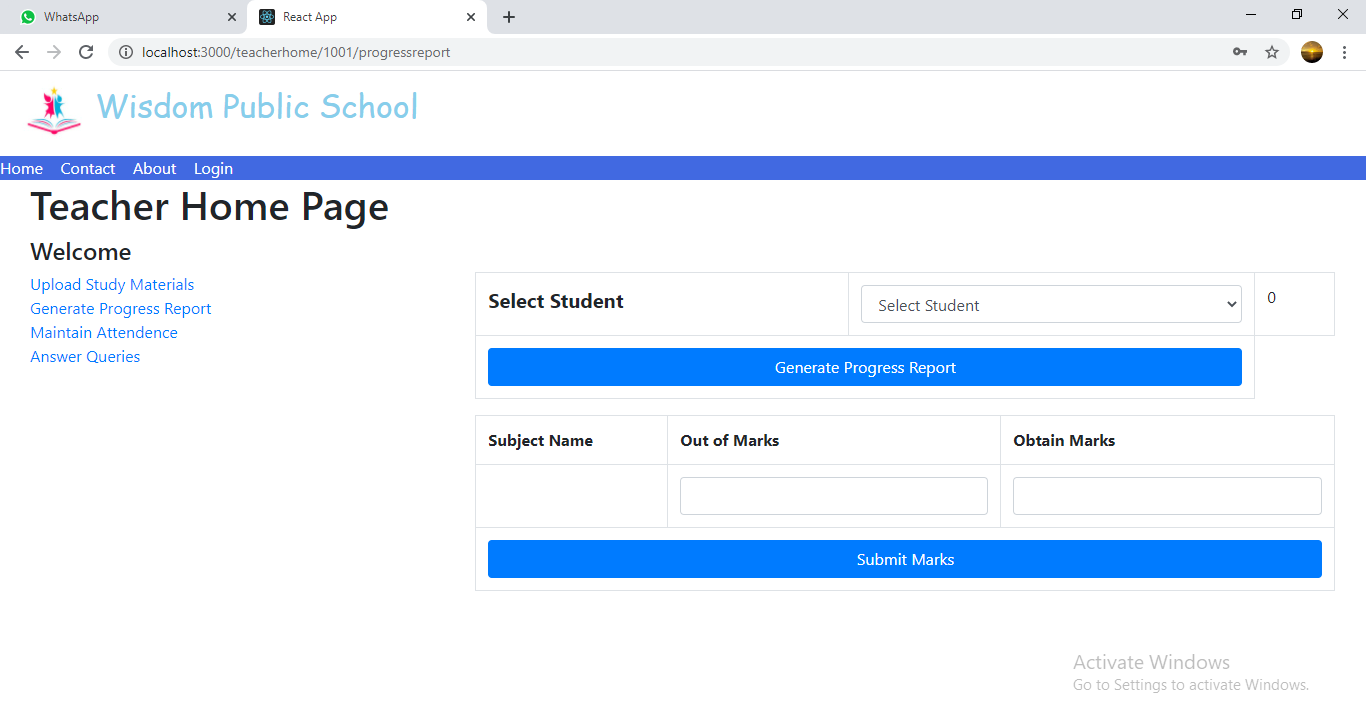


**8.6 Teacher Home Page**

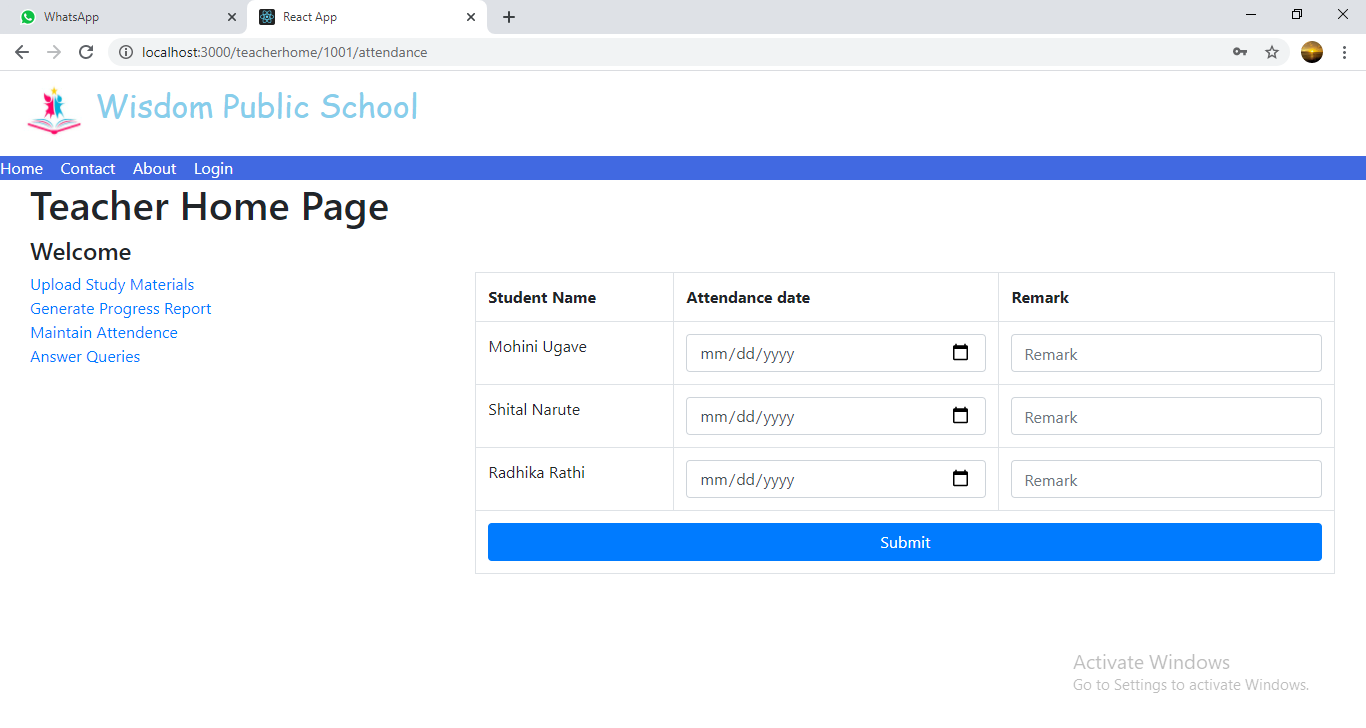
**8.6.1 Upload Study Material Page**

****

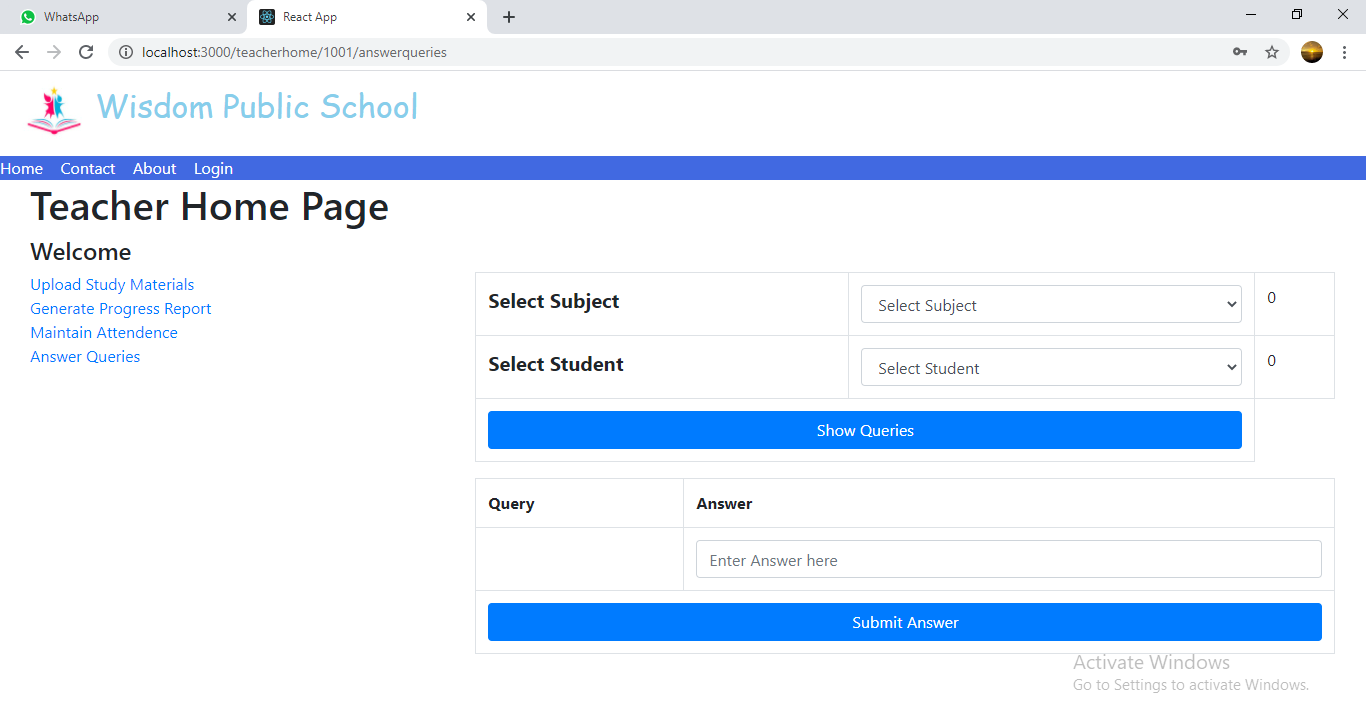
**8.6.2 Generate Progress Report Page**



**8.6.3 Maintain Attendance Page**

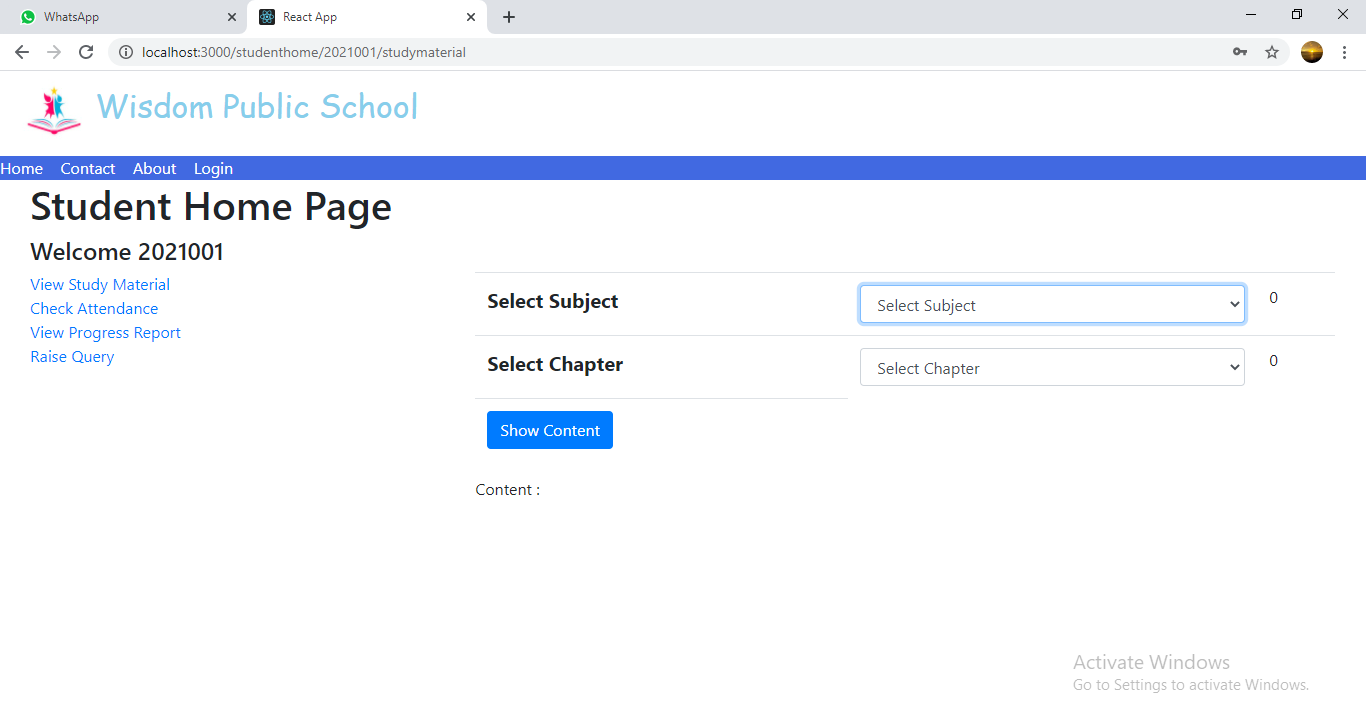


**8.6.4 Answer Queries Page**

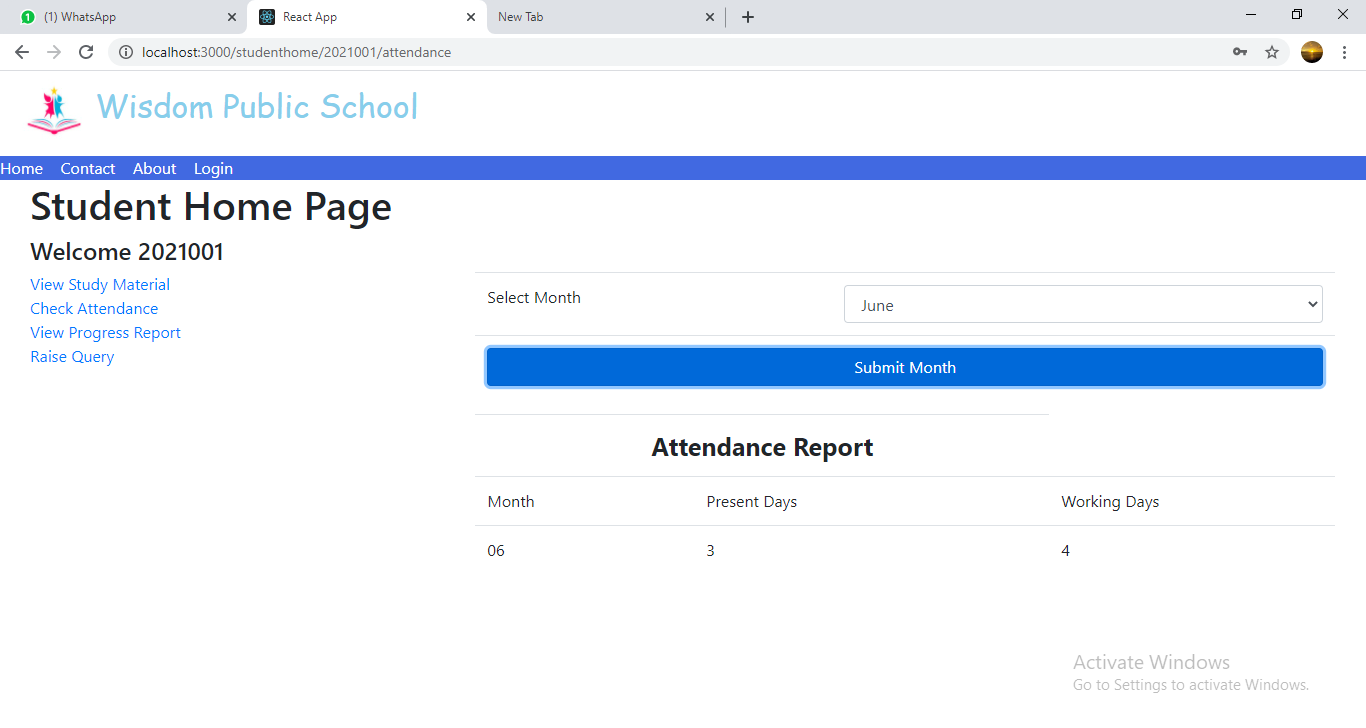


**8.7 Student Home Page**

**8.7.1 View Study Material Page**

****

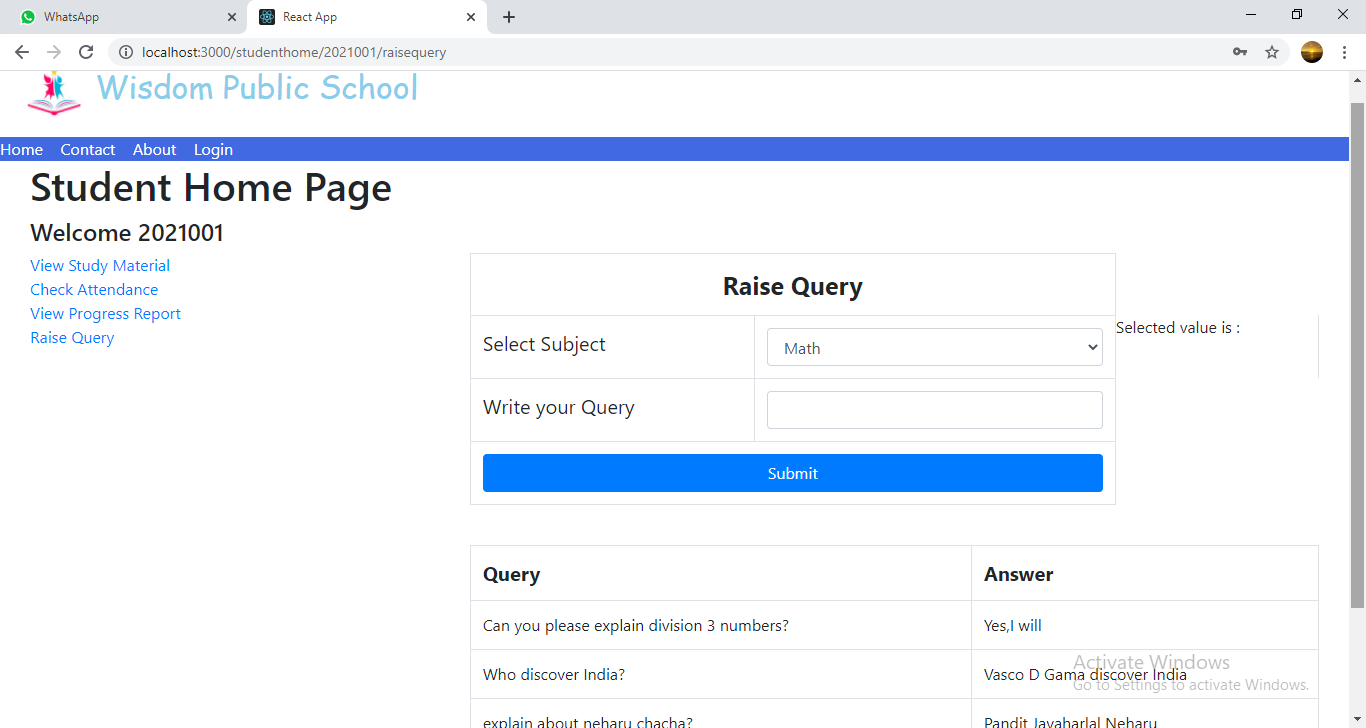
**8.7.2 Check Attendance Page**



**8.7.3 View Progress Report Page**



**8.7.4 Raise Query Page**



**9. CONCLUSION AND FUTURE SCOPE**

Paperless School Management System provides better education and platform to get connected with Teachers and Students efficiently.

Our System provides a very user-friendly platform where Student can easily read the study materials, check the attendance, raise the queries and view progress report, also Teacher will upload study materials, maintain attendance, answer queries and generate progress report.

Our system is aimed efficient management of various tasks like Generating time table for students on daily basis, Managing fees payment, Examination management. Digitally monitor daily activities along with managing all the resources and information on a single platform.