



Green University of Bangladesh

*Department of Computer Science and Engineering (CSE)
Semester: (Spring, Year: 2024), B.Sc. in CSE (Day)*

Student Portal

Course Title: Web Programming Lab

Course Code: CSE 302

Section: 221 D3

Students Details

Name	ID
Pritam Saha Turja	221002115

Submission Date: 24-12-2024

Course Teacher's Name: Mr. Kazi Hasnayeen Emad

[For teachers use only: **Don't write anything inside this box**]

<u>Lab Project Status</u>	
Marks:	Signature:
Comments:	Date:

Contents

1	Introduction	3
1.1	Overview	3
1.2	Motivation	3
1.3	Problem Definition	4
1.3.1	Problem Statement	4
1.3.2	Complex Engineering Problem	5
1.4	Design Goals/Objectives	5
1.5	Application	6
1.6	Conclusion	7
2	Design/Development/Implementation of the Project	8
2.1	Introduction	8
2.2	Project Details	8
2.2.1	Block Diagram	9
2.3	Implementation	10
2.3.1	The workflow	13
2.3.2	Tools and libraries	13
3	Performance Evaluation	14
3.1	Simulation Environment/ Simulation Procedure	14
3.2	Results Analysis/Testing	15
3.2.1	Homepage	15
3.2.2	Administration Side	16
3.2.3	Student Side	18
3.3	Results Overall Discussion	21
4	Conclusion	22
4.1	Discussion	22

4.2	Limitations	22
4.3	Scope of Future Work	23

Chapter 1

Introduction

1.1 Overview

The Student Portal Web Application is a complete platform that aims to simplify and improve students' academic experiences. This application has a straightforward and user-friendly design, and students may securely log in using a pre-saved username and password. Key features include class routine generation depending on enrolled courses, a dynamic CGPA calculator, and transcript generation to provide an accurate academic viewpoint. Students can also examine their exam results, check their billing history, and simply register for the next semester. Built with HTML, CSS, PHP and a MySQL database, the platform offers a strong and secure backend for efficient management of all functionalities. The system prioritizes flexibility by allowing all information, including course enrollments, outcomes, and billing data, to be updated in real time, assuring accuracy and usefulness.

This project promotes an integrated approach to student management by automating typical academic tasks while ensuring data integrity. Designed to satisfy the needs of educational institutions, the portal is flexible and customizable, making it suitable for a variety of academic systems. Its development serves as a practical implementation of modern web technologies and database management concepts, contributing to an optimal digital academic system.

1.2 Motivation

The reason for building the Student Portal Web Application derives from the growing need for digital academic administration solutions, as well as the desire to simplify students' academic experiences. In many educational institutions, handling student data, course registrations, outcomes, and billing can be a time-consuming and difficult procedure. These manual approaches are prone to mistakes, delays, and inefficiencies, which may harm a student's entire experience and progress. This project aims to address these issues by offering a single platform for automating essential academic activities, making them easier, quicker, and more easily available. By including functions such as class routine creation, CGPA calculation, result monitoring, and bill history review, the site enables students to manage their academic activities freely, creating a sense of control

and accountability. Also, the project gives students an actual chance to learn about and apply web development concepts including PHP, MySQL, HTML, and CSS in a real-world environment. It matches up with the larger purpose of helping to the digitization of education, which has become even more important in the context of modern educational practices.

Finally, the objective is to create a system that helps students while minimizing administrative responsibilities, resulting in a smooth, efficient, and user-friendly academic experience.

1.3 Problem Definition

1.3.1 Problem Statement

Many educational institutions depend on manual processes or divided systems to manage academic activities such as course registration, outcome monitoring, and financial transactions, which can be inefficient and costly. Students face difficulties in obtaining critical information such as class schedules, CGPA, and billing history, which may hinder academic performance and preparation. Administrators also struggle to keep correct and up-to-date records for different functions. The absence of a single system causes delays, confusion, and a bad user experience. This project aims to address these challenges by creating a Student Portal Web Application—a merged user-friendly platform that allows students to easily handle their academic demands while administrators benefit from simplified data management and real-time updates.

1.3.2 Complex Engineering Problem

Table 1.1: Summary of the attributes touched by the mentioned projects

Name of the P Attributes	Explain how to address
P1: Depth of knowledge required	HTML,CSS,PHP,MYSQL
P2: Range of conflicting requirements	To create a balanced and efficient Student portal ,overcoming these challenging desires requires careful evaluation,stakeholder interaction,process design, and testing processes.
P3: Depth of analysis required	System design requires an in-depth review of student portal tasks and user needs.
P4: Familiarity of issues	The project's goal is to simple student portal,which users should be aware with.
P5: Extent of applicable codes	The project involves writing,implementing, and maintaining code for Student portal tools.
P6: Extent of stakeholder involvement and conflicting requirements	Consider stakeholders different needs,such as user-friendliness and security
P7: Interdependence	Efficient Student portal required careful design and implementation because of their interdependence

1.4 Design Goals/Objectives

- **User-Friendly Interface:** Create an easy and appealing design that allows students to locate and access all skills.
- **Centralized System :** Create a single system that combines essential academic functions such as course enrollment, routine generation, outcome tracking, billing history, and CGPA calculation.
- **Real-Time Updates :** Ensure that the system offers real-time updates on course enrollments, grades, billing, and register.
- **Data Accuracy and Security:** Use accurate data validation techniques to verify accuracy and protect important student information through encryption and secure authentication systems.
- **Scalability:** Design the portal to allow for future additions, such as more academic features or interface with institutional systems.
- **Responsiveness:** To improve the user experience, ensure that the portal is available on a variety of platforms, such as PCs, tablets, and smartphones.

1.5 Application

The Student Portal Web Application is a useful resource for educational institutions like schools, colleges, and universities. It tackles various real-world school administration concerns by providing a single system for streamlining administrative operations and improving the student experience. The following are detailed real-world applications :

- **Simplified Student Academic Management:** Students frequently have confusion about their course timetables, particularly when dealing with many courses. The site allows them to create customized class routines based on their registered courses, resulting in greater clarity and time management. Compared to waiting for paper results, students can view their academic success online. The automatic CGPA calculator replaces manual calculations, decreasing errors and delivering more precise performance measures.
- **Seamless Registration Process:** The portal allows semester registration by allowing students to register for courses online. This avoids long lines and paperwork, resulting in a quick and immediate enrollment procedure.
- **Enhanced Financial Transparency:** By providing a full billing history, the site helps students in keeping track of their tuition costs, payments, and pending dues. This avoids misunderstandings and ensures financial transparency for students and families.
- **Real-Time Updates and Accessibility:** All aspects of the interface, including results, routines, and payment information, are updated in real time.
- **Digital Transformation in Education:** By digitization many academic activities, the site contributes to the global trend of digital transformation in education. It decreases dependency on paper-based systems, cuts cost, and supports sustainable behaviors.
- **Improved Student Engagement and Satisfaction:** Allowing students to conduct their academic activities freely develops a sense of control and ownership. This increases their commitment to their academic development and general happiness with the institution.

In summary, the Student Portal Web Application meets the needs of both students and administrators, resulting in a more efficient, transparent, and user-friendly academic environment. It bridges the gap between conventional educational practices and modern technology, making it a useful tool for schools committed to providing high-quality education.

1.6 Conclusion

In summary, the Student Portal Web Application is a comprehensive solution that improves academic management by automating critical procedures such as course registration, class routine generation, CGPA calculation, result tracking, and billing history. The portal is designed with user-friendly interfaces, safe data processing, and flexibility in mind, making it more efficient for both students and administrators. It gives students real-time access to their academic data while decreasing administrative workload, resulting in a more visible and efficient academic environment. The project makes an important contribution to the digital transformation of educational institutions by connecting traditional education techniques with new technology, delivering a smooth academic experience.

Chapter 2

Design/Development/Implementation of the Project

2.1 Introduction

The project covers the creation and implementation of a Student Portal System, which will improve academic management for both students and administrators. This system has many functions such as administration login, add students with required information ,course teacher and grade update, student login, profile viewing, result viewing, bill history tracking. The goal is to build an all-in-one platform that improves efficiency and accessibility to academic assignments. The project makes use of powerful technologies and frameworks to offer a consistent user experience, secure data management, and scalability for future improvements. It has user-friendly interfaces that relate to a wide audience and supports dynamic updating to keep up with changing academic needs.

This technology is going to significantly decrease manual administrative effort, promote flexibility in student information, and give students with a powerful tool for academic self-management.

2.2 Project Details

The Student Portal System is a complete web-based solution that simplifies academic management for students and administrators. The system contains essential features such as:

- Administrations login
- Add new students record
- Update and delete student record
- Teachers login
- Student login

- Profile viewing Result showing
- Billing history

The system has dynamic functionality that allows for improves and changes based on institutional requirements. It has a user-friendly interface and secure backend data management to assure dependability and efficiency. This project intends to decrease administrative costs, improve data accessibility, and enable students to manage their academic activities more efficiently.

2.2.1 Block Diagram

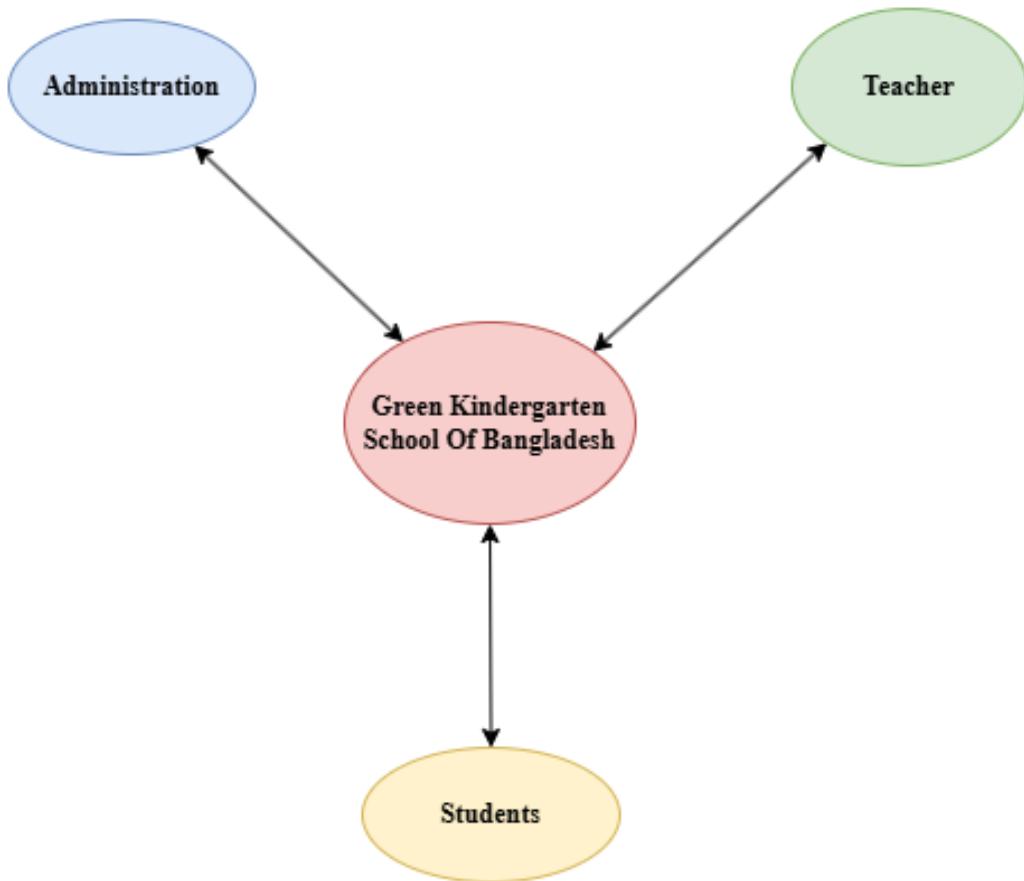
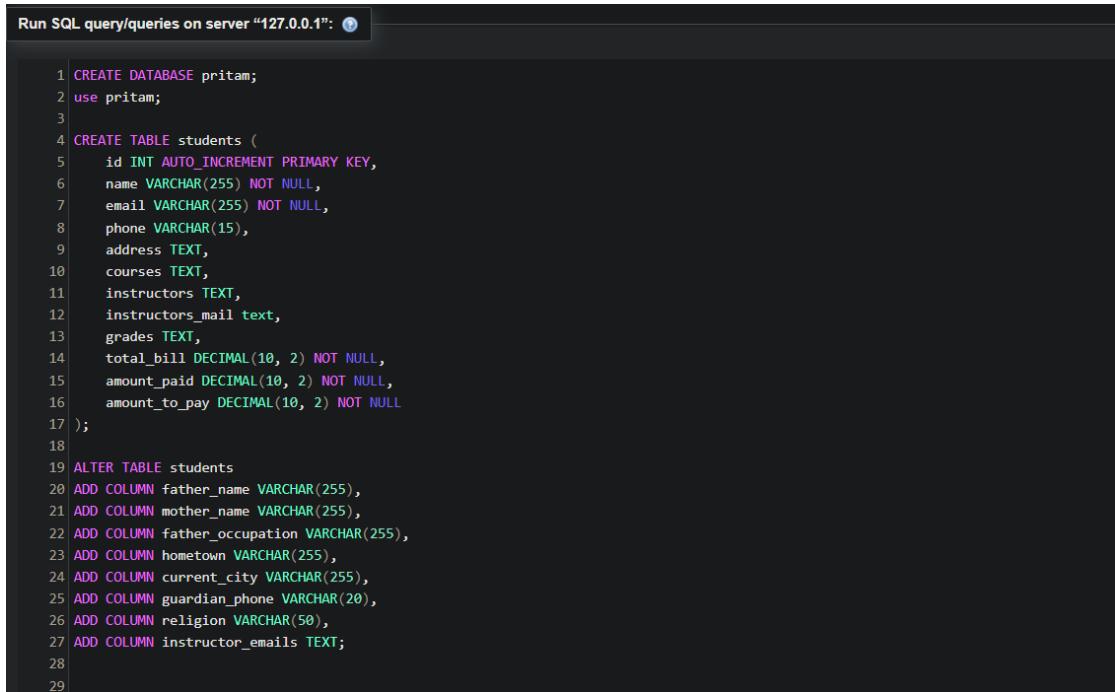


Figure 2.2.1.1: Block Diagram for Green kindergarten school web application

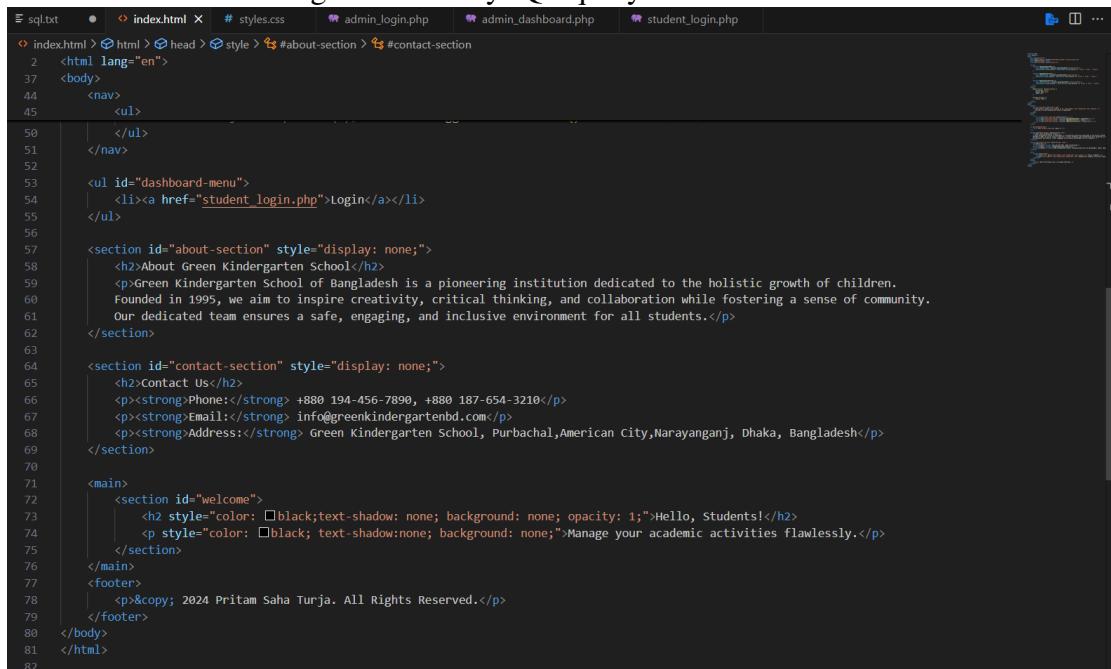
2.3 Implementation



The screenshot shows the MySQL Workbench interface with a SQL editor tab. The title bar says "Run SQL query/queries on server "127.0.0.1":". The code in the editor is as follows:

```
1 CREATE DATABASE pritam;
2 use pritam;
3
4 CREATE TABLE students (
5     id INT AUTO_INCREMENT PRIMARY KEY,
6     name VARCHAR(255) NOT NULL,
7     email VARCHAR(255) NOT NULL,
8     phone VARCHAR(15),
9     address TEXT,
10    courses TEXT,
11    instructors TEXT,
12    instructors_mail text,
13    grades TEXT,
14    total_bill DECIMAL(10, 2) NOT NULL,
15    amount_paid DECIMAL(10, 2) NOT NULL,
16    amount_to_pay DECIMAL(10, 2) NOT NULL
17 );
18
19 ALTER TABLE students
20 ADD COLUMN father_name VARCHAR(255),
21 ADD COLUMN mother_name VARCHAR(255),
22 ADD COLUMN father_occupation VARCHAR(255),
23 ADD COLUMN hometown VARCHAR(255),
24 ADD COLUMN current_city VARCHAR(255),
25 ADD COLUMN guardian_phone VARCHAR(20),
26 ADD COLUMN religion VARCHAR(50),
27 ADD COLUMN instructor_emails TEXT;
28
29
```

Figure 2.3.1: MySQL query for database



The screenshot shows a code editor with multiple tabs open. The active tab is "index.html". The code is as follows:

```
sql.txt index.html # styles.css admin_login.php admin_dashboard.php student_login.php
index.html > html > head > style > #about-section > #contact-section
1 <html lang="en">
2   <head>
3     <meta charset="UTF-8" />
4     <title>Green Kindergarten School</title>
5     <link href="styles.css" rel="stylesheet" />
6   </head>
7   <body>
8     <nav>
9       <ul>
10      <li><a href="#">Home</a></li>
11      <li><a href="#">About Us</a></li>
12      <li><a href="#">Contact Us</a></li>
13      <li><a href="#">Login</a></li>
14    </ul>
15  </nav>
16
17  <section id="about-section" style="display: none;">
18    <h2>About Green Kindergarten School</h2>
19    <p>Green Kindergarten School of Bangladesh is a pioneering institution dedicated to the holistic growth of children. Founded in 1995, we aim to inspire creativity, critical thinking, and collaboration while fostering a sense of community. Our dedicated team ensures a safe, engaging, and inclusive environment for all students.</p>
20  </section>
21
22  <section id="contact-section" style="display: none;">
23    <h2>Contact Us</h2>
24    <p><strong>Phone:</strong> +880 194-456-7890, +880 187-654-3210</p>
25    <p><strong>Email:</strong> info@greenkindergartenbd.com</p>
26    <p><strong>Address:</strong> Green Kindergarten School, Purbachal, American City, Narayanganj, Dhaka, Bangladesh</p>
27  </section>
28
29  <main>
30    <section id="welcome">
31      <h2 style="color: black; text-shadow: none; background: none; opacity: 1;">Hello, Students!</h2>
32      <p style="color: black; text-shadow: none; background: none;">Manage your academic activities flawlessly.</p>
33    </section>
34  </main>
35  <footer>
36    <p>&copy; 2024 Pritam Saha Turja. All Rights Reserved.</p>
37  </footer>
38
39 </body>
40 </html>
```

Figure 2.3.2: HTML code for homepage structure

A screenshot of a code editor window titled "styles.css". The editor shows CSS code for styling a homepage. The code includes rules for the body, header, nav ul, and heading elements. It also includes a style for an image within the header and a class for error messages. The editor interface includes tabs for other files like "index.html", "admin_login.php", "admin_dashboard.php", and "student_login.php". The status bar at the bottom shows "Ln 1, Col 1" and "Spaces: 4".

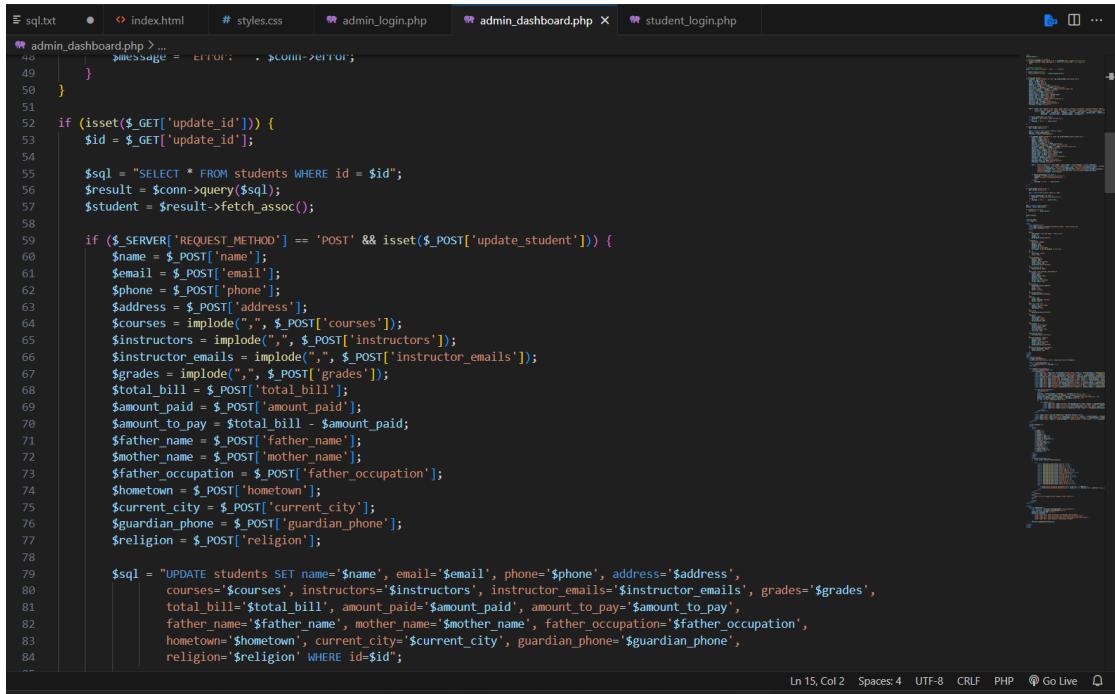
```
1 body {  
2     font-family: 'Times New Roman', Times, serif;  
3     margin: 0;  
4     padding: 0;  
5     height: 100vh;  
6     background-image: url('background.jpg');  
7     background-size: cover;  
8     background-repeat: no-repeat;  
9     background-position: center;  
10    background-attachment: fixed;  
11 }  
12  
13  
14 header {  
15     text-align: center;  
16     background: none;  
17     padding: 20px 0;  
18 }  
19  
20 header img {  
21     max-width: 150px;  
22     border-radius: 50%;  
23 }  
24  
25  
26 #heading {  
27     margin: 0;  
28     color: white;  
29 }  
30  
31  
32 nav ul {  
33     list-style: none;  
34     padding: 0;  
35     margin: 20px 0;  
36     text-align: center;  
37 }
```

Figure 2.3.3: CSS code for homepage styling

A screenshot of a code editor window titled "admin_login.php". The editor shows PHP code for an admin login page. It includes HTML for a login form with fields for username and password, and a submit button. It also includes CSS styles for the login container and a button, and a PHP block for handling errors. The editor interface includes tabs for "index.html", "styles.css", "admin_login.php", "admin_dashboard.php", and "student_login.php". The status bar at the bottom shows "Ln 9, Col 5" and "Spaces: 4".

```
30 <html lang="en">  
31     <head>  
32         <style>  
33             .login-container button {  
34                 border-radius: 5px;  
35                 cursor: pointer;  
36             }  
37             .login-container button:hover {  
38                 background: #0056b3;  
39             }  
40             .error {  
41                 color: red;  
42                 margin-bottom: 10px;  
43             }  
44         </style>  
45     </head>  
46  
47     <body>  
48         <div class="login-container">  
49             <h1>Admin Login</h1>  
50             <?php if (isset($error)): ?>  
51                 <p class="error"> <?= $error ?> </p>  
52             <?php endif; ?>  
53             <form method="POST">  
54                 <label for="username">Username:</label>  
55                 <input type="text" id="username" name="username" required>  
56  
57                 <label for="password">Password:</label>  
58                 <input type="password" id="password" name="password" required>  
59  
60                 <button type="submit">Login</button>  
61             </form>  
62         </div>  
63     </body>
```

Figure 2.3.4: Admin Login



```

sql.txt index.html # styles.css admin_login.php admin_dashboard.php student_login.php

48     $msg = "ERROR: " . $conn->error;
49 }
50 }

51 if (isset($_GET['update_id'])) {
52     $id = $_GET['update_id'];
53 }

54 $sql = "SELECT * FROM students WHERE id = $id";
55 $result = $conn->query($sql);
56 $student = $result->fetch_assoc();

57 if ($_SERVER['REQUEST_METHOD'] == 'POST' && isset($_POST['update_student'])) {
58     $name = $_POST['name'];
59     $email = $_POST['email'];
60     $phone = $_POST['phone'];
61     $address = $_POST['address'];
62     $courses = implode(",", $_POST['courses']);
63     $instructors = implode(",", $_POST['instructors']);
64     $instructor_emails = implode(",", $_POST['instructor_emails']);
65     $grades = implode("", $_POST['grades']);
66     $total_bill = $_POST['total_bill'];
67     $amount_paid = $_POST['amount_paid'];
68     $amount_to_pay = $total_bill - $amount_paid;
69     $father_name = $_POST['father_name'];
70     $mother_name = $_POST['mother_name'];
71     $father_occupation = $_POST['father_occupation'];
72     $hometown = $_POST['hometown'];
73     $current_city = $_POST['current_city'];
74     $guardian_phone = $_POST['guardian_phone'];
75     $religion = $_POST['religion'];

76     $sql = "UPDATE students SET name='$name', email='$email', phone='$phone', address='$address',
77             courses='$courses', instructors='$instructors', instructor_emails='$instructor_emails', grades='$grades',
78             total_bill='$total_bill', amount_paid='$amount_paid', amount_to_pay='$amount_to_pay',
79             father_name='$father_name', mother_name='$mother_name', father_occupation='$father_occupation',
80             hometown='$hometown', current_city='$current_city', guardian_phone='$guardian_phone',
81             religion='$religion' WHERE id=$id";
82 }

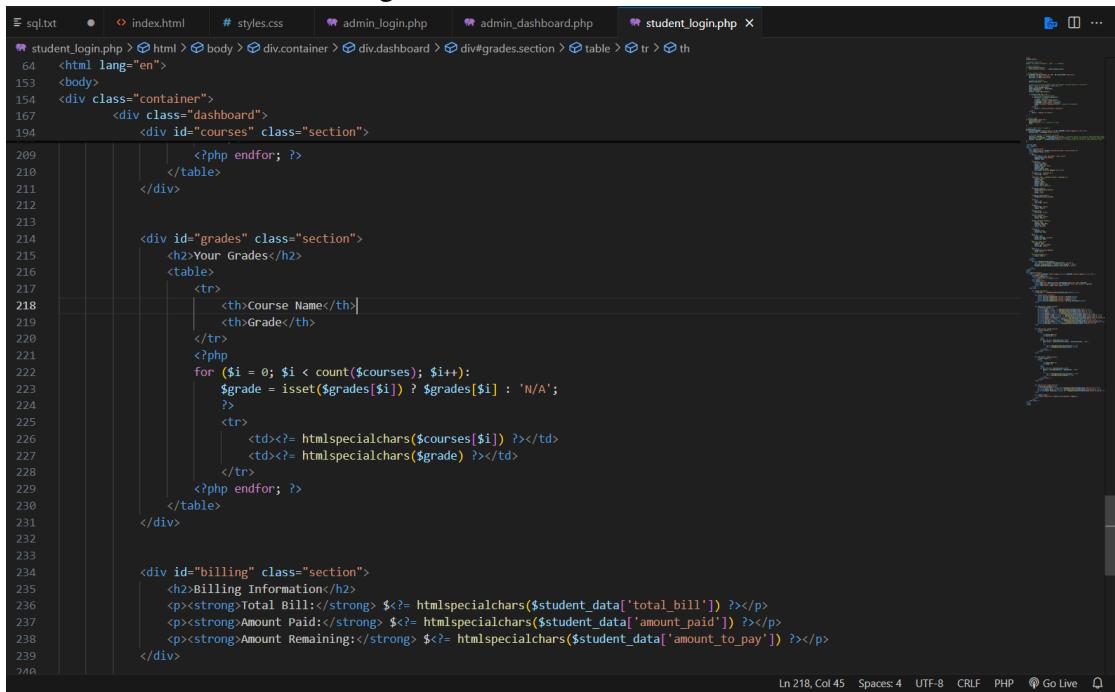
83 }

84

```

Ln 15, Col 2 Spaces: 4 UTF-8 CRLF PHP Go Live

Figure 2.3.5: Admin Dashboard



```

sql.txt index.html # styles.css admin_login.php admin_dashboard.php student_login.php

student_login.php > html > body > div.container > div.dashboard > div#grades.section > table > tr > th
64 <html lang="en">
65 <body>
66     <div class="container">
67         <div class="dashboard">
68             <div id="courses" class="section">
69                 <?php endforeach; ?>
70             </div>
71         </div>
72     </div>

73     <div id="grades" class="section">
74         <h2>Your Grades</h2>
75         <table>
76             <tr>
77                 <th>Course Name</th>
78                 <th>Grade</th>
79             </tr>
80             <?php
81             for ($i = 0; $i < count($courses); $i++) {
82                 $grade = isset($grades[$i]) ? $grades[$i] : 'N/A';
83             }
84             <?php endforeach; ?>
85         </table>
86     </div>

87     <div id="billing" class="section">
88         <h2>Billing Information</h2>
89         <p><strong>Total Bill:</strong> $<?= htmlspecialchars($student_data['total_bill']) ?></p>
90         <p><strong>Amount Paid:</strong> $<?= htmlspecialchars($student_data['amount_paid']) ?></p>
91         <p><strong>Amount Remaining:</strong> $<?= htmlspecialchars($student_data['amount_to_pay']) ?></p>
92     </div>
93 
```

Ln 218, Col 45 Spaces: 4 UTF-8 CRLF PHP Go Live

Figure 2.3.6: Student Login and dashboard

2.3.1 The workflow

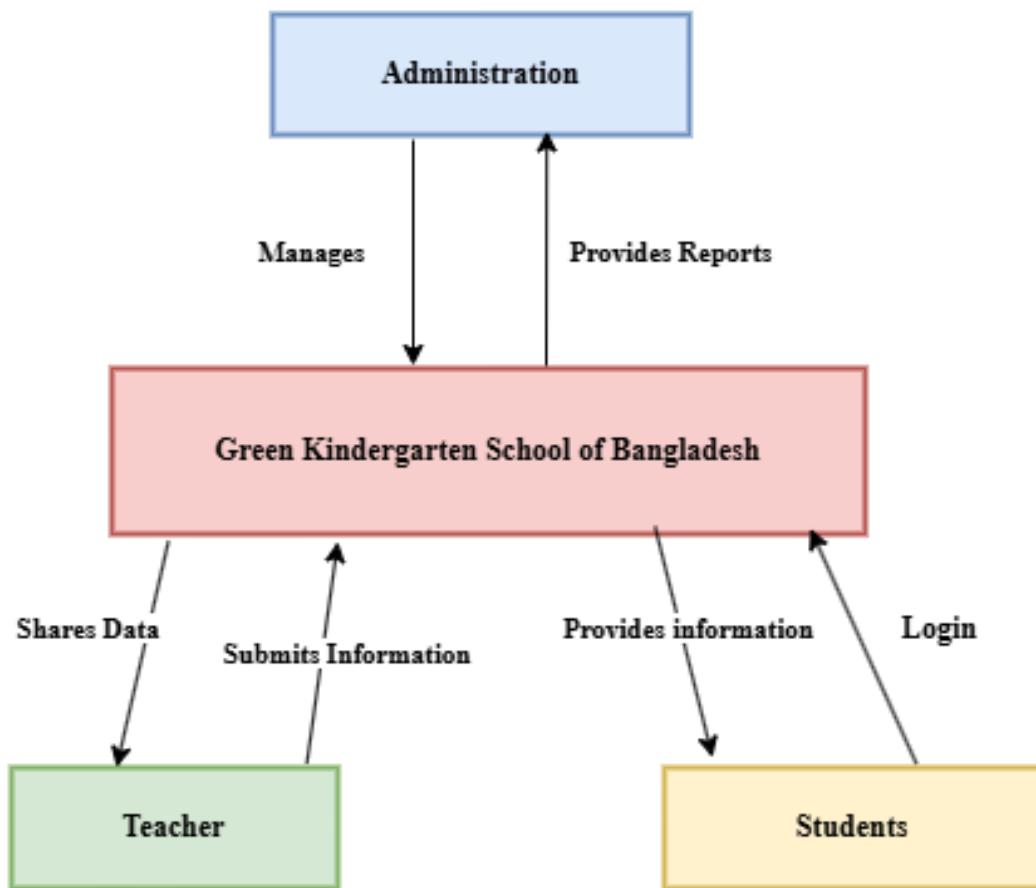


Figure 2.3.1.1: DFD level zero diagram for the web application

2.3.2 Tools and libraries

Tools and technologies used to develop this project are given below ;

- Language used: HTML,CSS,JS,PHP,MySQL
- Tools: VS code, XAMPP
- Documentation: Latex(Overleaf)

Chapter 3

Performance Evaluation

3.1 Simulation Environment/ Simulation Procedure

To simulate and evaluate the outcomes of the Student Portal System, the experimental setup involves configuring a development and testing environment that supports both the frontend and backend requirements of the project. Below is a detailed outline of the setup:

Development Tools:

Code Editor: Visual Studio Code

Web Server: XAMPP for local server hosting

Database Server: MySQL for data storage

Languages and Frameworks:

Frontend: HTML, CSS, JavaScript

Backend: PHP for server-side logic

Database Queries: MySQL queries

3.2 Results Analysis/Testing

3.2.1 Homepage



Figure 3.2.1.1: Homepage



Figure 3.2.1.2: About Section



Figure 3.2.1.3: Contact Information

3.2.2 Administration Side

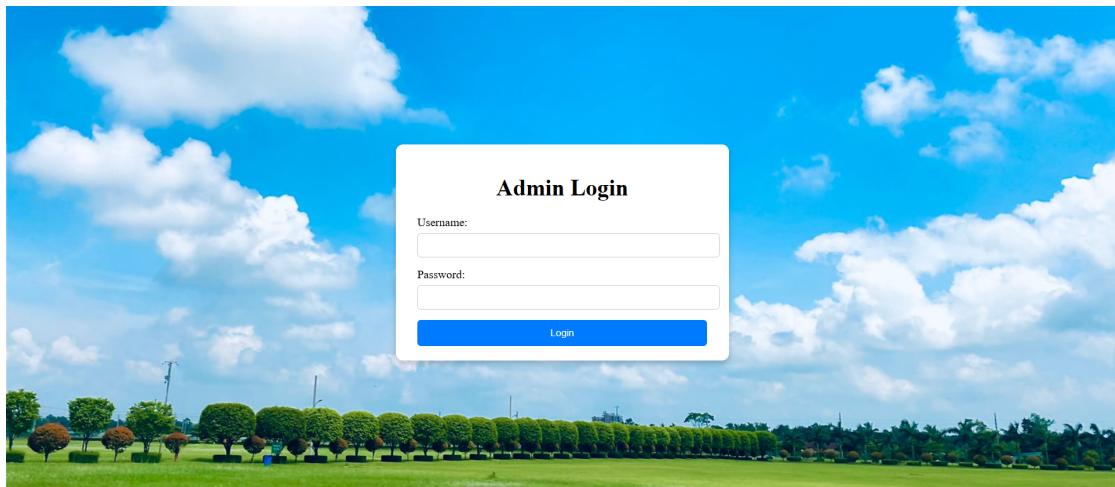


Figure 3.2.2.1: Admin/teachers Login

A detailed screenshot of the "Admin Login" form. It shows the title "Admin Login" at the top. Below it are two input fields: one for "Username" containing "kaziemad" and another for "Password" containing ".....". A large blue "Login" button is positioned at the bottom of the form.

Figure 3.2.2.2: Admin login with username and password

Add or Update Student

Pranto Bapary
prantobapary1@gmail.com
01647185767
Mirpur 10
Md. Sajjad Hossain
Rabyea Hossain
Businessman
Sherpur
Dhaka
01703713943
Muslim

Figure 3.2.2.3: Add new student information

Courses

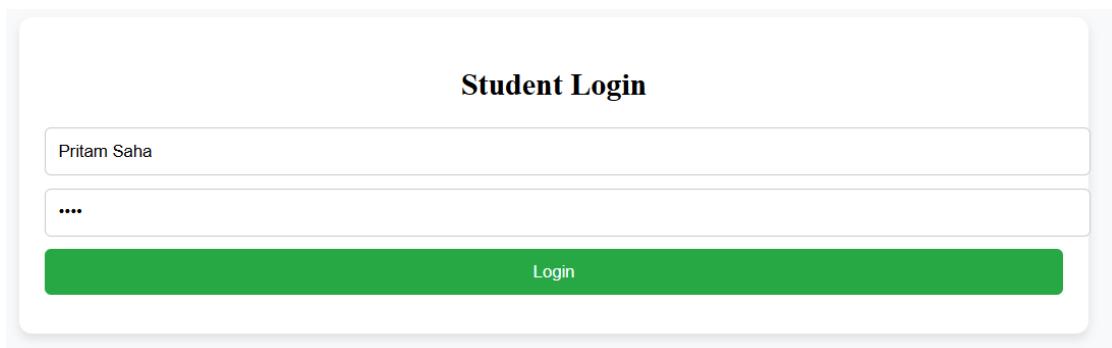
Bangla	
Kamrul Hasan	
kamrulhasan@green.edu.bd	
A+	
English	
Saiful Islam	
saifulislam@green.edu.bd	
A+	
Math	
Mr. Kazi Hasnayeen Emad	
kaziemad@green.edu.bd	
A+	
Add More Courses	
23000.00	
21000	
Update Student	

Figure 3.2.2.4: Update student information

All Students															
Name	Email	Phone	Address	Father's Name	Mother's Name	Father's Occupation	Hometown	Current City	Guardian Phone	Religion	Courses	Total Bill	Amount Paid	Amount To Pay	Actions
Pritam Saha	pritamsahaturja.11@gmail.com	01703713943	Dhaka,Bangladesh	Uttam Kumar Saha	Mita Rani Saha	Businessman	Dhaka	Dhaka	01703713943	Hinduism	Bangla	12300.00	1200.00	11100.00	Update Delete
Maruf Sarker	marufsackar@gmail.com	01798024366	Motijheel	Kaamrul Hasan	Rokeya Hasan	Service	Norail	Dhaka	01558861228	Muslim	English,Bangladesh Studies	20000.00	14500.00	5500.00	Update Delete
Pranto Bapary	prantobapary1@gmail.com	01647-185767	Mirpur 10	Md. Sajjad Hossain	Rabyea Hossain	Businessman	Sherpur	Dhaka	01703713943	Muslim	Bangla,English,Math	23000.00	21000.00	2000.00	Update Delete
Arif Billah	arifbillah@gmail.com	01987645321	Kanchan	Md. Akram Hossain	Ms. Asma Begum	Businessman	Baarishal	Narayanganj	01876543212	Muslim	English,English,Math	30000.00	24000.00	6000.00	Update Delete

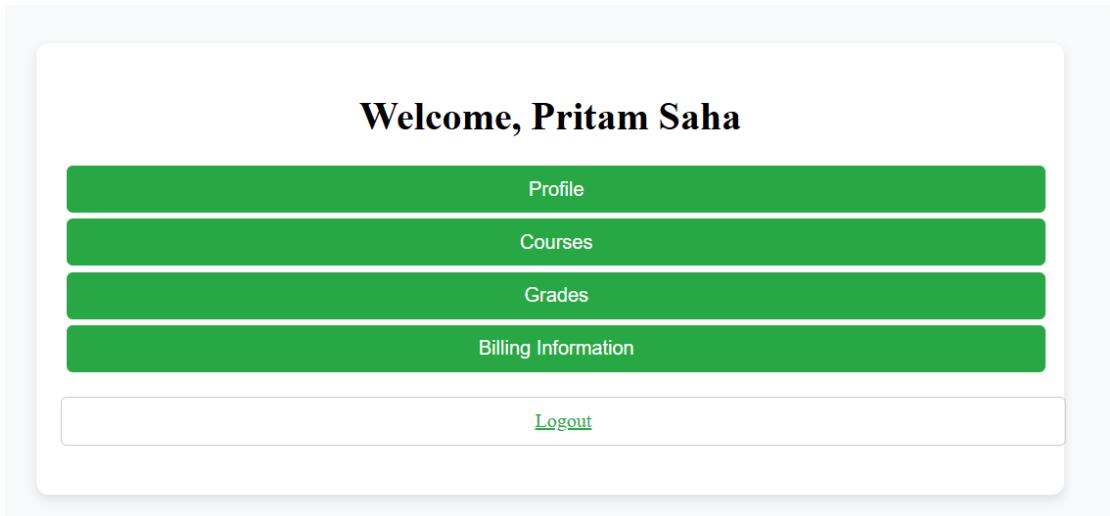
Figure 3.2.2.5: All student record from the database

3.2.3 Student Side



The form is titled "Student Login". It contains two input fields: one for "Username" containing "Pritam Saha" and another for "Password" containing "....". Below the fields is a green "Login" button.

Figure 3.2.3.1: Student login with username and password



The dashboard is titled "Welcome, Pritam Saha". It features a navigation bar with four green buttons: "Profile", "Courses", "Grades", and "Billing Information". At the bottom is a white "Logout" button.

Figure 3.2.3.2: Student Dashboard

Welcome, Pritam Saha



Profile Details

Name: Pritam Saha

Email: pritamsahaturja.11@gmail.com

Phone: 01703713943

Address: Dhaka,Bangladesh

Father's Name: Uttam Kumar Saha

Mother's Name: Mita Rani Saha

Father's Occupation: Businessman

Hometown: Dhaka

Current City: Dhaka

Guardian Phone: 01703713943

Religion: Hinduism

[Logout](#)

Figure 3.2.3.3: Showing student profile (General information)

The screenshot shows a user interface for a student's profile. At the top, it says "Welcome, Pritam Saha". Below that is a horizontal menu bar with four green buttons: "Profile", "Courses", "Grades", and "Billing Information". Underneath the menu, the section title "Your Courses" is displayed. A table follows, with columns "Course Name" and "Instructor". The data in the table is as follows:

Course Name	Instructor
Bangla	Kamrul Hasan
Math	Mr. Kazi Hasnayeen Emad
English	Saiful Islam

At the bottom of the page is a "Logout" button.

Figure 3.2.3.5: Showing course list and instructor for that courses for the specific student)

The screenshot shows a user interface for a student's profile. At the top, it says "Welcome, Pritam Saha". Below that is a horizontal menu bar with four green buttons: "Profile", "Courses", "Grades", and "Billing Information". Underneath the menu, the section title "Your Grades" is displayed. A table follows, with columns "Course Name" and "Grade". The data in the table is as follows:

Course Name	Grade
Bangla	A+
Math	A+
English	A+

At the bottom of the page is a "Logout" button.

Figure 3.2.3.6: Showing Grades for the courses

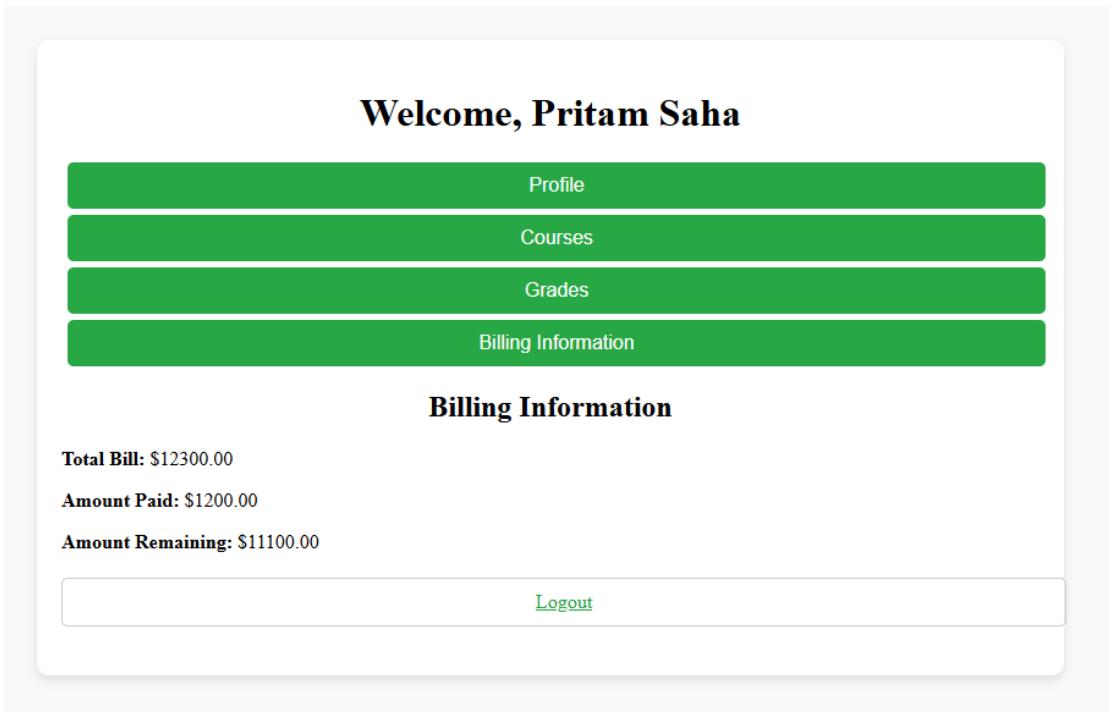


Figure 3.2.3.7: Showing Billing history for that specific Student

3.3 Results Overall Discussion

The results analysis/testing show that the "Green Kindergarten School of Bangladesh" system achieves its primary goals of improving administrative, teacher, and student interactions. Key results include:

- **Features Functionality:** Core functions such as user authentication, data management, and report production function properly, delivering critical services to the institution.
- **Data Management:** The installation of a database guarantees that information is efficiently stored and retrieved.
- **Usability:** User interface and user experience were most likely assessed depending on how easily administrators, teachers, and students connected with the program.
- **Performance:** The system's ability to manage multiple users, response speeds, and database efficiency may have been evaluated. Scalability or performance measures could have a significant impact on the outcomes.

Chapter 4

Conclusion

4.1 Discussion

The findings and observations presented in this chapter concentrate on the functional and usability testing of the "Green Kindergarten School of Bangladesh" system, highlighting its success in achieving fundamental requirements. The system was tested for critical functionality such as user identification, data management, and report production to ensure operational reliability. Observations indicate a user-friendly design and effective management of administrative, teacher, and student interactions. Scalability, comprehensive error handling, and speed optimization are all areas that could be improved for future enhancements. Overall, the chapter comments on how the project accomplishes its basic objectives while highlighting areas for refinement to improve its ability to adapt and flexibility.

4.2 Limitations

- **Security Concerns:** Without appropriate encryption and security measures, possible holes in user authentication and data storage may disclose important information.
- **Dependency on Manual Input:** The system rely heavily on manual data entry, increasing the chances of human error.
- **Database Constraints:** The database structure may not be designed for flexibility or complicated queries, resulting in possible difficulties as data volume grows.
- **Separate authentication for different teachers:** The system has a drawback that is there is no separate authentication for several teachers login , which may cause sometimes some vulnerability if there arise lack of ethics.

4.3 Scope of Future Work

- **Separate authentication for different teachers:** In future I will try to integrate more advanced security for separate authentication for separate teachers , that will increase the security of the system more.
- **Language Localization:**Add international functionality to make the portal more accessible to users with varied language backgrounds.
- **Analytics and Reporting:**Introduce accurate data for administrators and teachers to collect information into student performance, attendance, and financial trends.
- **Integration with External Systems:**Expand the portal's functionality by integrating third-party solutions such as Learning Management Systems (LMS), payment gateways etc.
- **Additional Features:**Expanding the system to support new user roles or functionality, such as automatic notifications or engagement with other educational resources.

References

<https://www.w3schools.com/html/>

<https://www.w3schools.com/css/>

<https://www.w3schools.com/js/>

<https://www.w3schools.com/php/>

<https://www.youtube.com/@anisul-islam>

<https://www.w3schools.com/MySQL/default.asp>