CHAPTER 1

INTRODUCTION

The Student Result Management System (SRMS) is a web-based application designed to simplify and streamline the process of managing student examination results within educational institutions. By providing a user-friendly platform for administrators, teachers, and students, SRMS aims to enhance transparency, accessibility, and efficiency in result management processes. This report provides an overview of the SRMS project, including its purpose, objectives, scope, and organization overview. Additionally, it outlines the system analysis, design, and implementation details, along with conclusions and references.

1.1 PURPOSE:

The Student Result Management System (SRMS) is designed to simplify how schools and colleges handle exam results. It acts like a digital assistant that helps organize and manage student scores from tests and exams. The main goal is to ensure accuracy, save time, and make it easy for everyone to access their results whenever they need them.

1.2 OBJECTIVE:

The big aim of SRMS is to create a user-friendly system that benefits everyone involved the education process. Teachers can input and manage results efficiently, students can access their scores easily, and administrators can oversee everything smoothly. By using SRMS, schools can streamline their result management process, reduce errors, and focus more on helping students succeed in their studies.

1.3 ORGANIZATION OVERVIEW

SRMS is meant for all types of educational institutions, from small schools to large universities. It serves as a central platform where teachers, students, and administrators can interact with exam results and related information.

Whether it's a primary school or a higher education institution, SRMS can adapt to meet their specific needs and requirements.

1.4 SCOPE OF THE PROJECT

The scope of the Student Result Management System (SRMS) project encompasses a wide range of functionalities aimed at improving the efficiency and effectiveness of result management processes within educational institutions. In addition to handling exam results, SRMS includes the following key features:

- Student Record Management: SRMS allows administrators to maintain comprehensive records of students, including personal information, contact details, academic history, and any other relevant data. This feature enables schools to easily track student progress and demographics.

- Semester Management: SRMS facilitates the management of academic semesters, including scheduling, course offerings, and academic calendars. Administrators can create, edit, and update semester information as needed, ensuring accurate and up-to-date scheduling for students and faculty.

- Subject Management: SRMS provides tools for managing academic subjects, including subject names, codes, descriptions, and prerequisites. This feature allows administrators to organize and categorize subjects according to departmental requirements and curriculum standards.

- User Authentication and Access Control: SRMS includes robust user authentication mechanisms to ensure secure access to the system. Administrators, teachers, and students are assigned specific roles and permissions based on their responsibilities and requirements, allowing for fine-grained access control and data security.

CHAPTER 2

SYSTEM ANALYSIS

2.1 REQUIREMENT SPECIFICATION:

The SRS deals with the requirement of the proposed system. It describes what the proposed System should do without describing how the System will do it. The basic objective of SRS is to specify the important requirements of the proposed system that are gathered during the system analysis. To satisfy the basic goals, an SRS should have some desirable characteristics: correctness, completeness, unambiguous, verifiable, modifiable and traceable, which we have throughout during the process in entirely.

2.2 SYSTEM STUDY

Understanding the existing processes and identifying areas for improvement is crucial for developing an effective Student Result Management System (SRMS). System study involves analyzing the current methods of managing student examination results and assessing the challenges faced by educational institutions. The following steps are involved in the system study phase of the SRMS project:

- Data gathering
- Study of existing system
- Analyzing problem
- Studying various documents
- Feasibility study for further improvements

During the initial study phase, the project team gathers information about the specific needs and challenges faced by educational institutions in managing student examination results. By studying the existing system and analyzing its shortcomings, the project team gains insights into the requirements and opportunities for improvement.

This information serves as the foundation for designing and developing a robust and user-friendly SRMS that meets the needs of stakeholders and enhances the efficiency and effectiveness of result management processes.

2.3 FUNCTION REQUIREMENT SPECIFICATIONS

The user screen be informative web page with different links and pop-ups which is simple and easy to understand all the services provided by the system.

Hardware Requirements:

Processor Speed	2GHz
Processor	Intel Pentium 4 or higher processor
Memory Size	1 GB RAM (minimum)
Storage	80GB Hard Disk (minimum)

Software Requirements:

Front End	PHP, HTML, CSS, JAVASCRIPT
Back End	MYSQL(XAMPP)
Operating System	WINDOWS 10 AND ANY OTHER COMPATIBLE OS

Communication Interface:

Database Connection, PHP.

CHAPTER 3

SYSTEM DESIGN

The most creative and challenging face of the system development is System Design. It provides the understanding and procedural details necessary for the logical and physical stages of development. In designing a new system, the system analyst must have clear understanding of how the output is to be produced and in what format. Second, input data and master files have to be designed to meet the requirements of the proposed output. The operational phases are handled through program construction and testing.

3.1 TABLE DESIGN

3.1.1 ADMIN (ADMIN_ID, USERNAME, PASSWORD)

FIELD NAME	DATA TYPE/SIZE	CONSTRAINT
ADMIN_ID	INT (50)	PRIMARY KEY
USERNAME	VARCHAR (100)	NOT NULL
PASSWORD	VARCHAR (60)	NOT NULL

3.1.2 BRANCH (BRANCH_ID, BRANCH)

FIELD NAME	DATA TYPE/SIZE	CONSTRAINT
BRANCH_ID	INT(50)	PRIMARY KEY
BRANCH	VARCHAR(100)	NOT NULL

3.1.3 SEMESTER (SEM_ID, SEMESTER)

FIELD NAME	DATA TYPE/SIZE	CONSTRAINT
SEM_ID	INT(9)	PRIMARY KEY
SEMESTER	INT(100)	NOT NULL

3.1.4 SUBJECTS (SUBJ_ID, SUBJ_NAME, SUBJ_CODE, STATUS)

FIELD NAME	DATA TYPE/SIZE	CONSTRAINT
SUBJ_ID	INT(200)	PRIMARY KEY
SUBJ_NAME	VARCHAR(200)	NOT NULL
SUBJ_CODE	VARCHAR(100)	NOT NULL
STATUS	INT(1)	NOT NULL

3.1.5 SUBJECT_COMB (COMB_ID, BRANCH_ID, SEM_ID, SUBJ_ID, STATUS)

FIELD NAME	DATA TYPE/SIZE	CONSTRAINT
COMB_ID	INT(200)	PRIMARY KEY
BRANCH_ID	INT(50)	NOT NULL
SEM_ID	INT(9)	NOT NULL
SUBJ_ID	INT(200)	NOT NULL
STATUS	INT(1)	NOT NULL

3.1.6 STUDENT (REG_ID, NAME, ROLL_NO, EMAIL, GENDER, DOB, BRANCH_ID, SEM_ID, REG_DATE, STATUS)

FIELD NAME	DATA TYPE/SIZE	CONSTRAINT
REG_ID	INT(225)	PRIMARY KEY
NAME	VARCHAR(200)	NOT NULL
ROLL_NO	INT(160)	NOT NULL
EMAIL	VARCHAR(100)	NOT NULL
GENDER	VARCHAR(50)	NOT NULL
DOB	VARCHAR(100)	NOT NULL
BRANCH_ID	INT(100)	NOT NULL
SEM_ID	INT(8)	NOT NULL
REG_DATE	TIMESTAMP	NOT NULL
STATUS	INT(1)	NOT NULL

3.1.7 RESULT (RESULT_ID, ROLL_NO, BRANCH_ID, SEM_ID, SUBJ_ID, MARKS)

FIELD NAME	DATA TYPE/SIZE	CONSTRAINT
RESULT_ID	INT(200)	PRIMARY KEY
ROLL_NO	INT(200)	NOT NULL
BRANCH_ID	INT(50)	NOT NULL
SEM_ID	INT(9)	NOT NULL
SUBJ_ID	INT(200)	NOT NULL
MARKS	INT(110)	NOT NULL

3.2 ENTITY RELATIONSHIP DIAGRAM

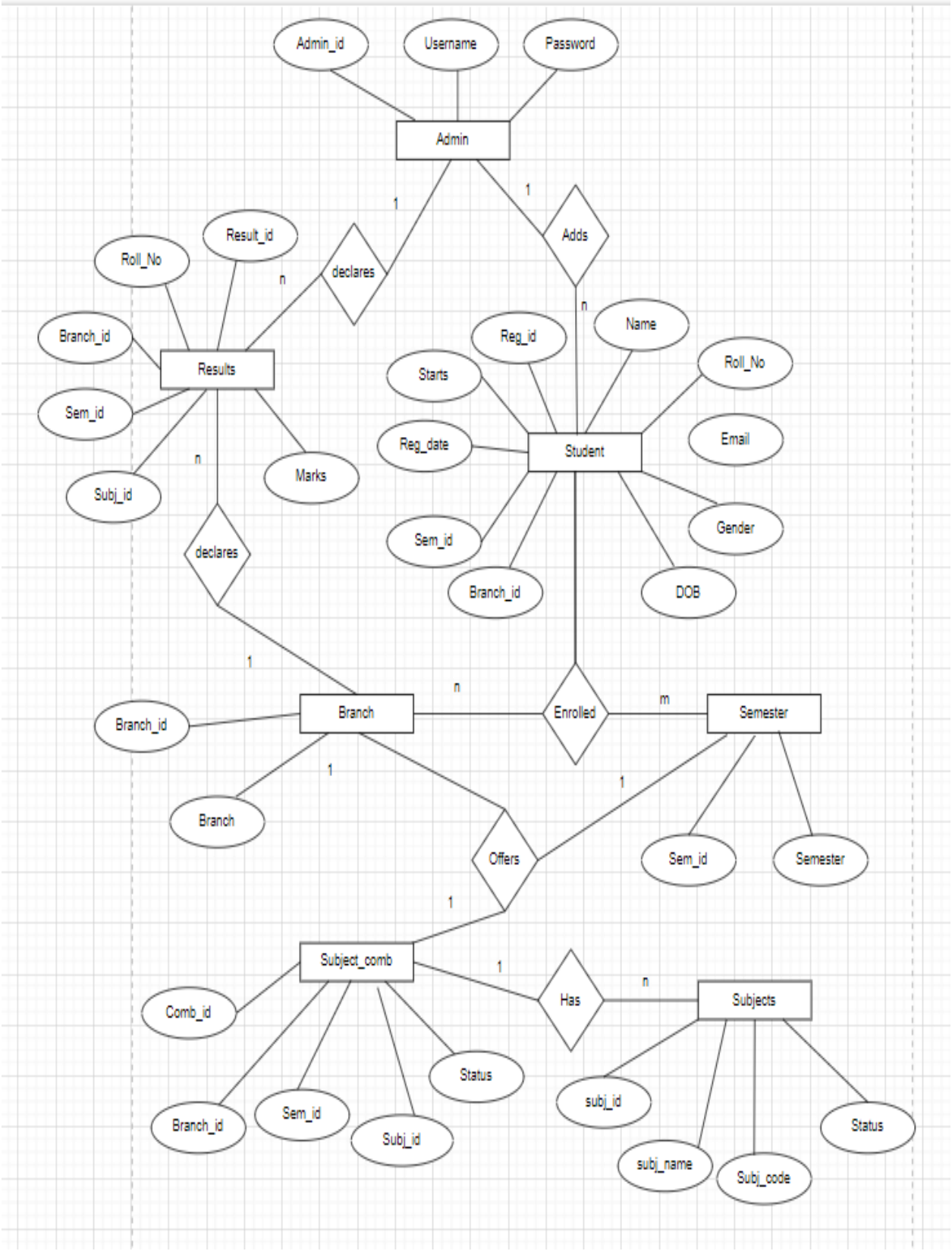
A set of primary component are identified for the E-R Diagram: data object, attributes, relationships and various type indicators. The primary purpose of the E-R Diagram is to represent data object and their relationships. Data objects: A data objects is a representation of almost any composite information that must be understood by software. Composite information refers to something that has a number of different properties or attributes.

Attributes: Attributes defines the properties of a data object and take on one of three different characteristics. They can be used to (1) name an instance of the data object, (2) describe the instance, or (3) make reference to another instance in another table. In addition, one or more of the attributes must be defined as an identifier that is, the identifier attribute becomes a "key" when we want to find an instance of the data object.

Relationship: Relationships indicate the manner in which data objects are "connected" to one another.

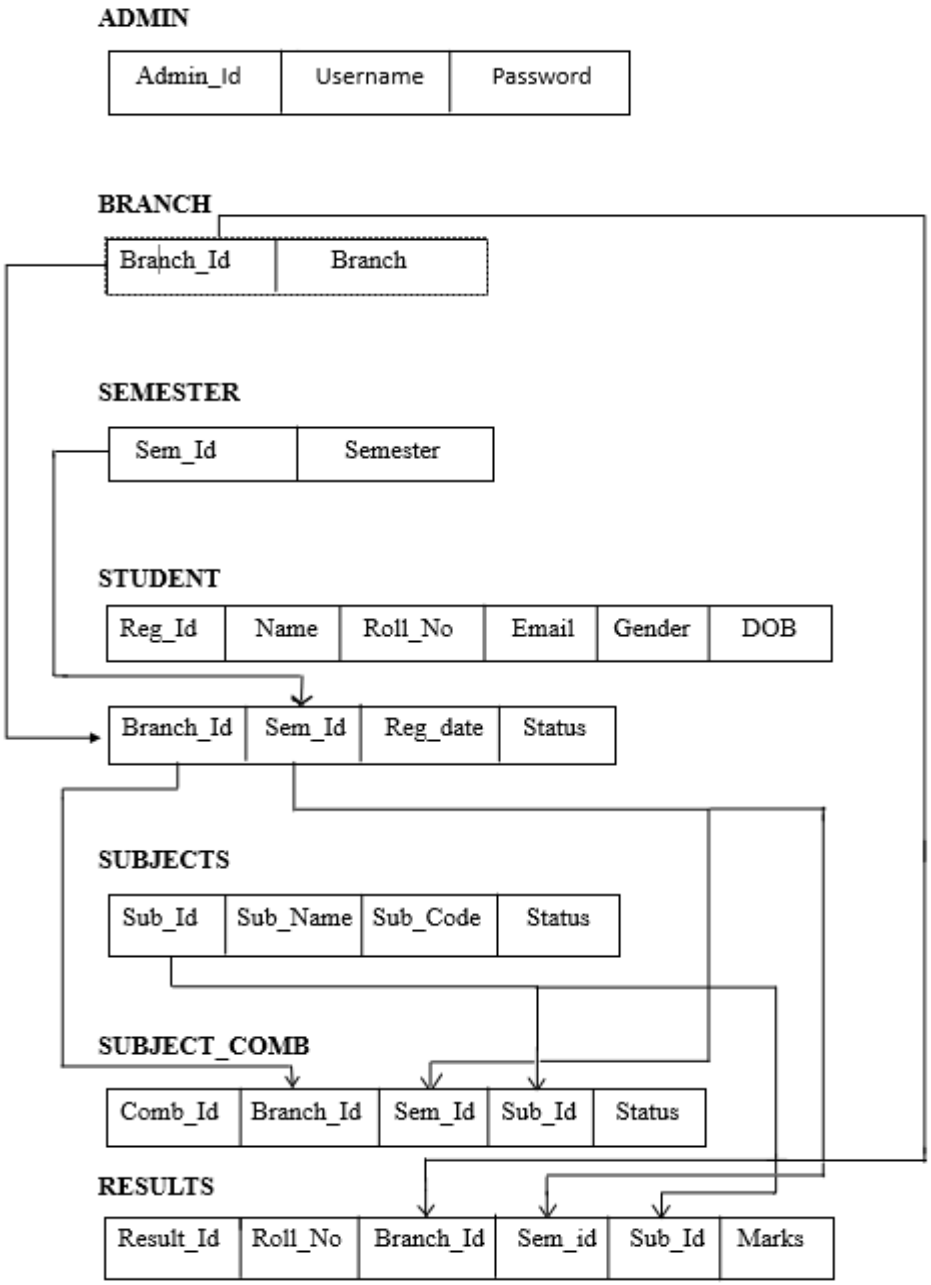
Cardinality: The data model must be capable of representing the number of occurrences objects in a given relationship. Tillman defines the cardinality of an object/relationship pair in the following manner: "Cardinality is the specification of the number of occurrences of one [object] that can be related to the number of occurrences of another [object]. Cardinality is usually expressed as simply 'one' and 'many', two [objects] can be as:

- One-to-One(1:1): An occurrence of [object] 'A' can relate to one and only one occurrence of [object] 'B', and an occurrence of 'B' can relate to only one occurrence of 'A'.
- One-to-Many (1:N): One occurrence of [object] 'A' can relate to one or many occurrences of [object] 'B', but an occurrence of 'B' can relate to only one occurrence of [object] 'A'.
- Many-to-Many(M:N): An occurrence of [object] 'A' can relate to one or more occurrence of 'B', while an occurrence of 'B' can relate to one or more occurrences of 'A' .


Admin_id
Username
Password
Admin
1
1
Adds
declares
Result_id
Roll_No
n
n
Branch_id
Results
Sem_id
Subj_id
n
Marks
declares
Reg_id
Name
Starts
Roll_No
Reg_date
Student
Email
Sem_id
Gender
Branch_id
DOB
1
n
m
Branch_id
Branch
Enrolled
Semester
1
1
Branch
Offers
Sem_id
Semester
1
Subject_comb
1
n
Has
Subjects
Comb_id
Status
Branch_id
Sem_id
Subj_id
subj_id
Status
subj_name
Subj_code

3.3 SCHEMA DIAGRAM

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data. A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams. It's the database designers who design the schema to help programmers understand the database and make it useful.



CHAPTER 4

IMPLEMENTATION

4.1 CODE(NAV.PHP):

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
 <link rel="stylesheet" href="css/nav.css">
</head>
<body>

<div class="navbar">
  <a href="dashboard.php">Dashboard | </a>
  <!-- <a href="#news">News</a> -->

  <div class="dropdown">
    <button class="dropbtn">Students
      <i class="fa fa-caret-down"></i>
    </button>
    <div class="dropdown-content">
      <a href="add-student.php" style="font-size : 20px">Add Students</a>
```

```
    <a href="manage-students.php" style="font-size : 20px">Manage Students</a>
    <!-- <a href="#">Link 3</a> -->
   </div>
  </div>
  <div class="dropdown">
   <button class="dropbtn">Branch
    <i class="fa fa-caret-down"></i>
   </button>
   <div class="dropdown-content">
    <a href="add-branch.php" style="font-size : 20px">Add branch</a>
    <a href="manage-branch.php" style="font-size : 20px">Manage branch</a>
    <!-- <a href="#">Link 3</a> -->
   </div>
  </div>
  <div class="dropdown">
   <button class="dropbtn">Semester
    <i class="fa fa-caret-down"></i>
   </button>
   <div class="dropdown-content">
    <a href="add-semester.php" style="font-size : 20px">Add semester</a>
    <a href="manage-sem.php" style="font-size : 20px">Manage semester</a>
    <!-- <a href="#">Link 3</a> -->
```

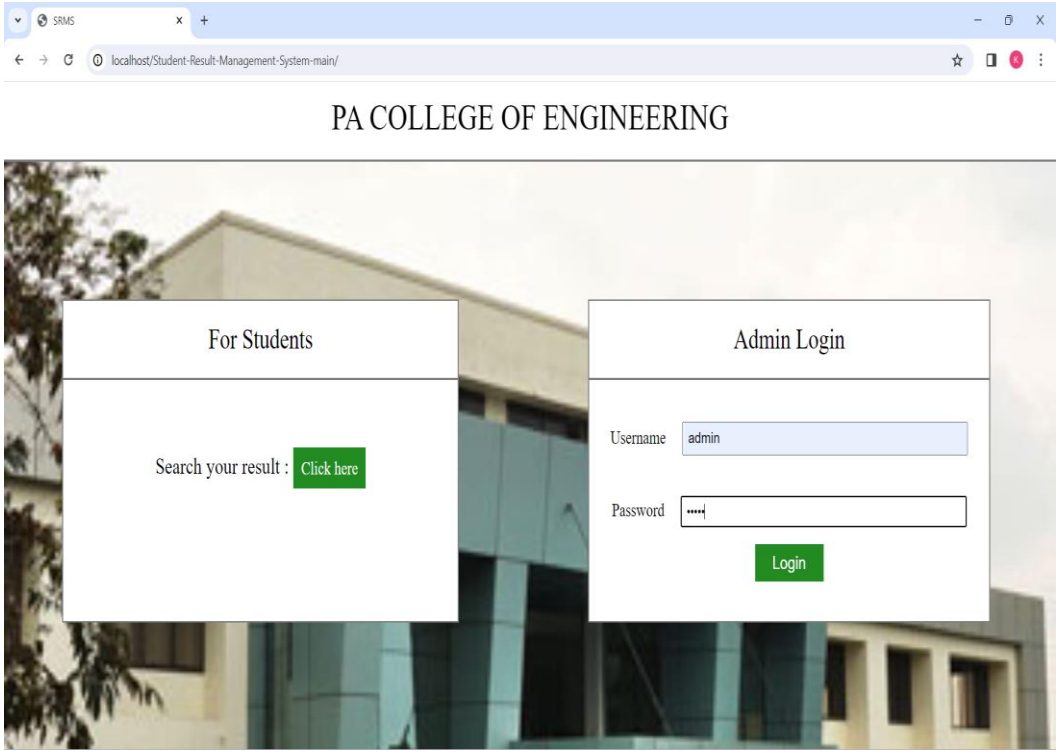
```
  </div>
 </div>
 <div class="dropdown">
  <button class="dropbtn">Subjects
   <i class="fa fa-caret-down"></i>
  </button>
  <div class="dropdown-content">
   <a href="add-subjects.php" style="font-size : 20px">Add Subjects</a>
   <a href="manage-subjects.php" style="font-size : 20px">Manage Subjects</a>
   <a href="add-subjcombo.php" style="font-size : 20px">Add Subject Combination</a>
   <a href="manage-subjcomb.php" style="font-size : 20px">Manage Subject Combination</a>
   <!-- <a href="#">Link 3</a> -->
  </div>
 </div>
 <div class="dropdown">
  <button class="dropbtn">Result
   <i class="fa fa-caret-down"></i>
  </button>
  <div class="dropdown-content">
   <a href="add-results.php" style="font-size : 20px">Add result</a>
   <a href="manage-results.php" style="font-size : 20px">Manage results</a>
```

```
    <!-- <a href="#">Link 3</a> -->
   </div>
  </div>
  <a href="change-password.php"style="font-size : 23px" >Admin Change Password </a>
  <a href="logout.php"style="font-size : 23px; float:right; margin-right : 10px" ><i class="fa fa-sign-out"></i> Logout </a>
</div>
</body>
</html>
```

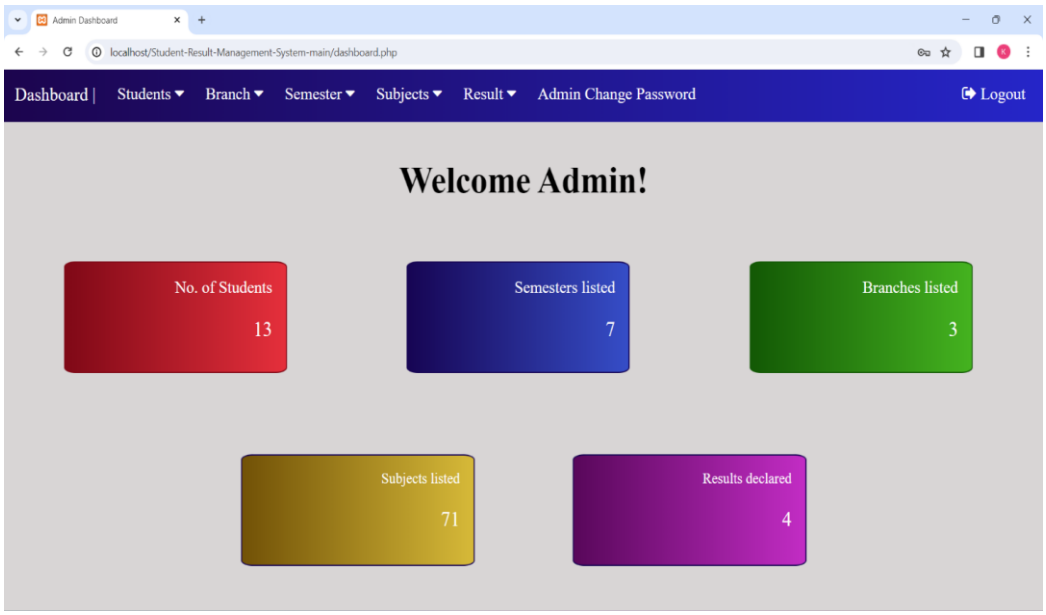
CHAPTER 5

SNAPSHOTS

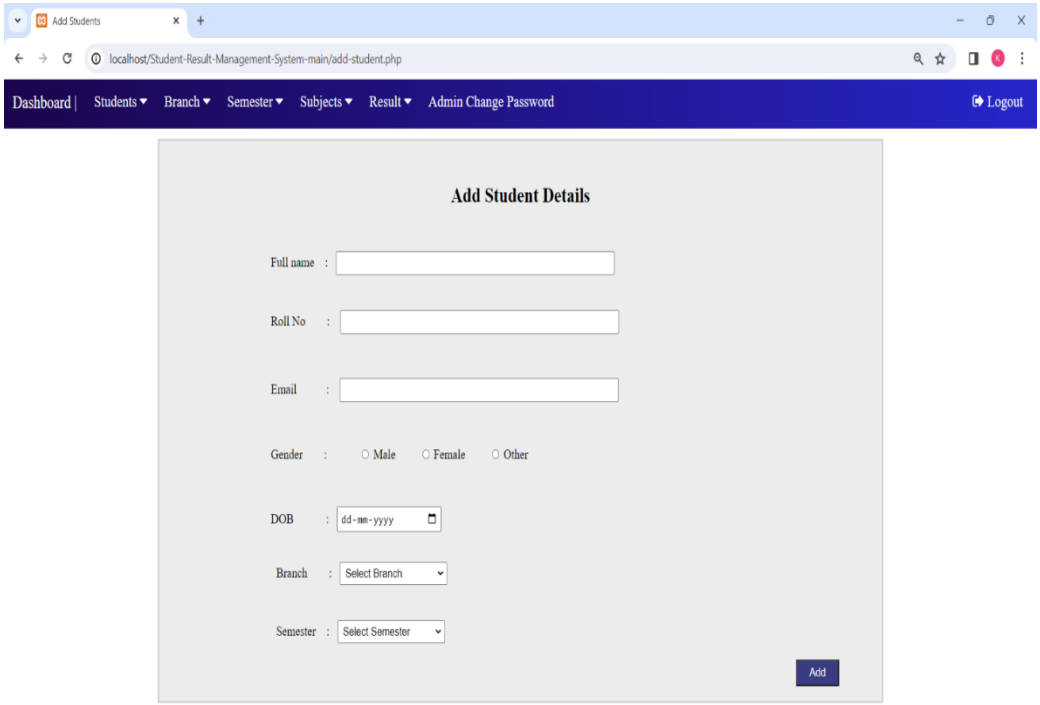
Home page: students can only search their results whereas admin can login with username and password and redirects to dashboard



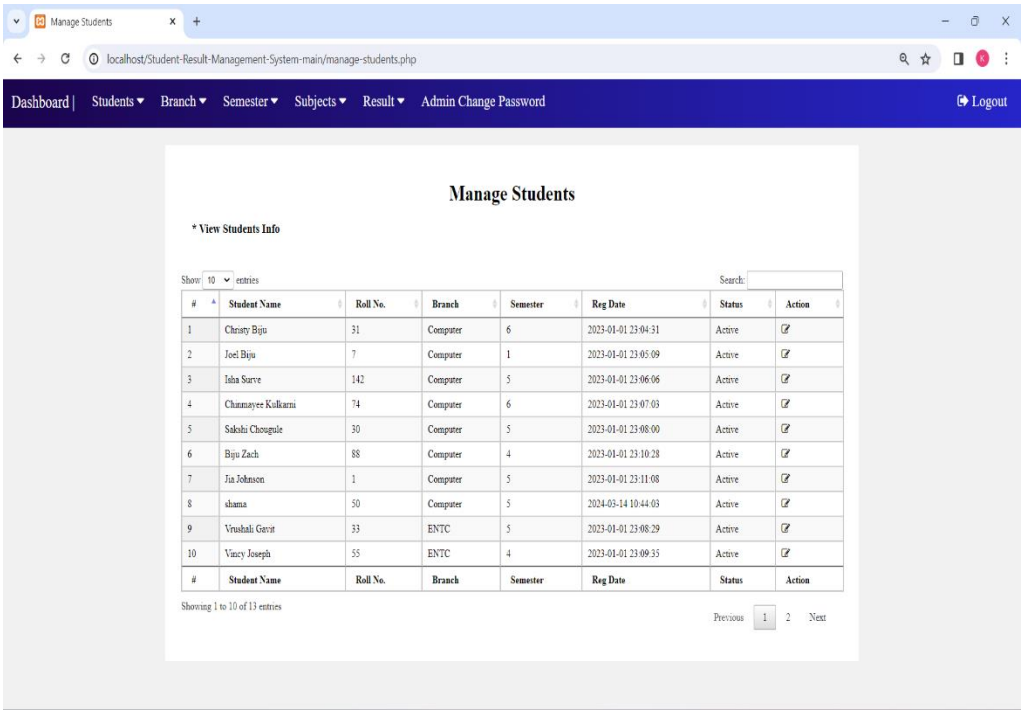
Dashboard: Admin can add, edit or view student data



Add Students: admin can enroll a student by filling the details of student

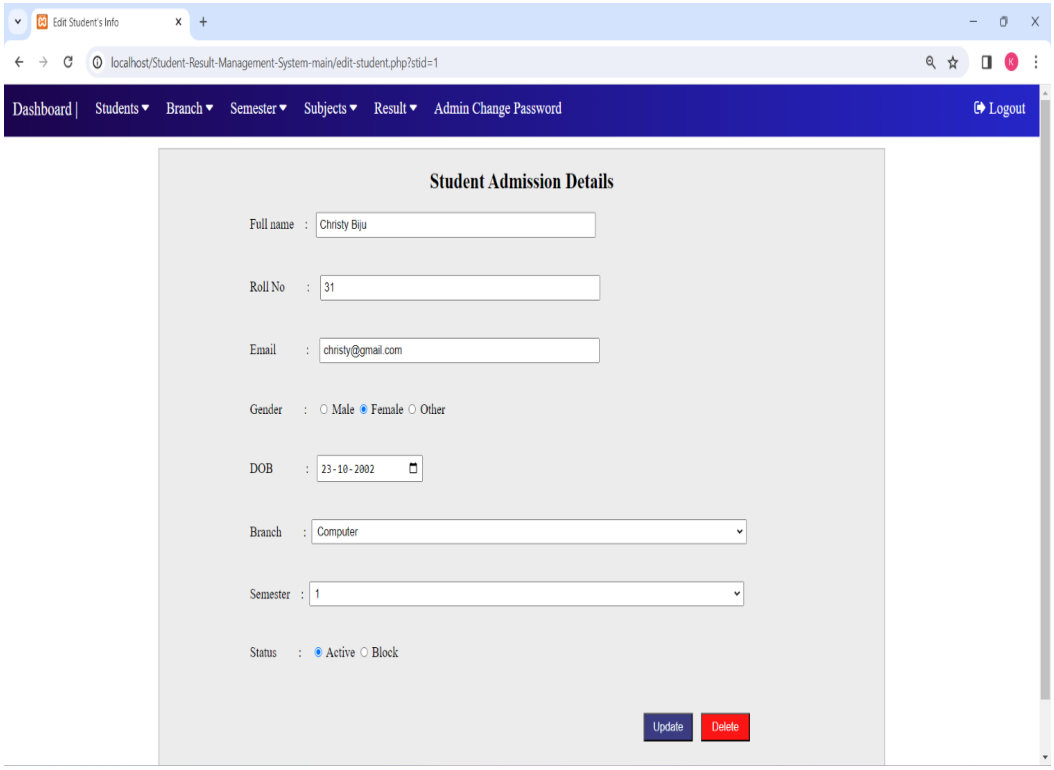


Manage Students: Admin can view all the enrolled students and also can edit or delete any student record

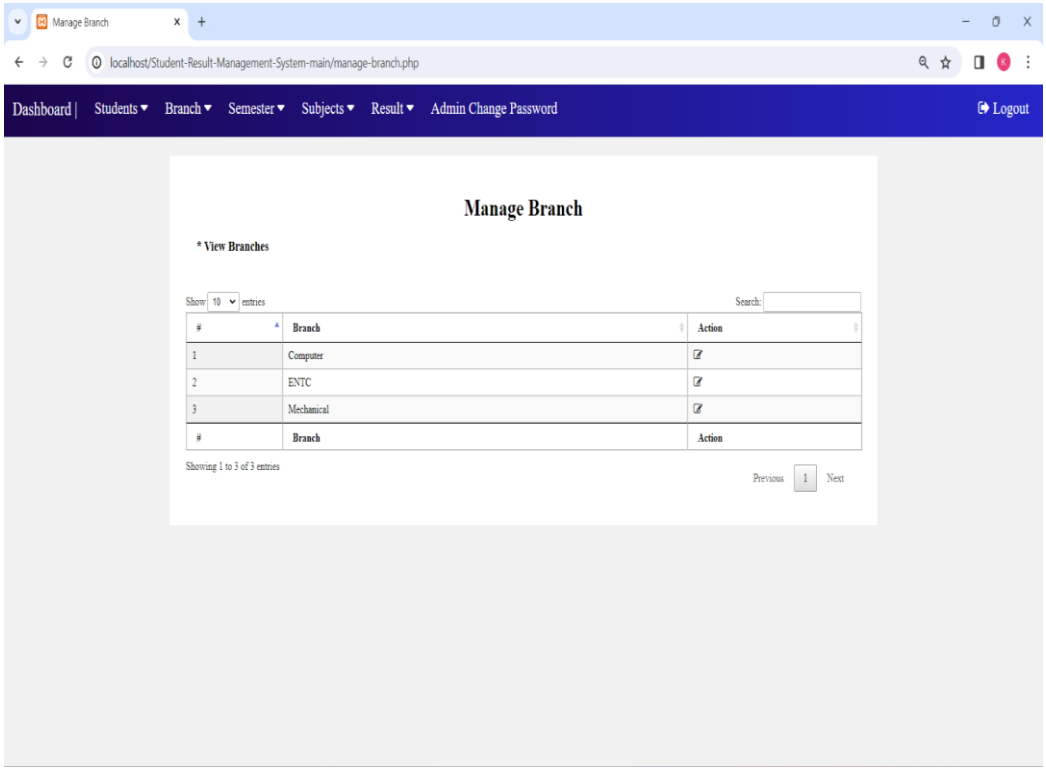


\

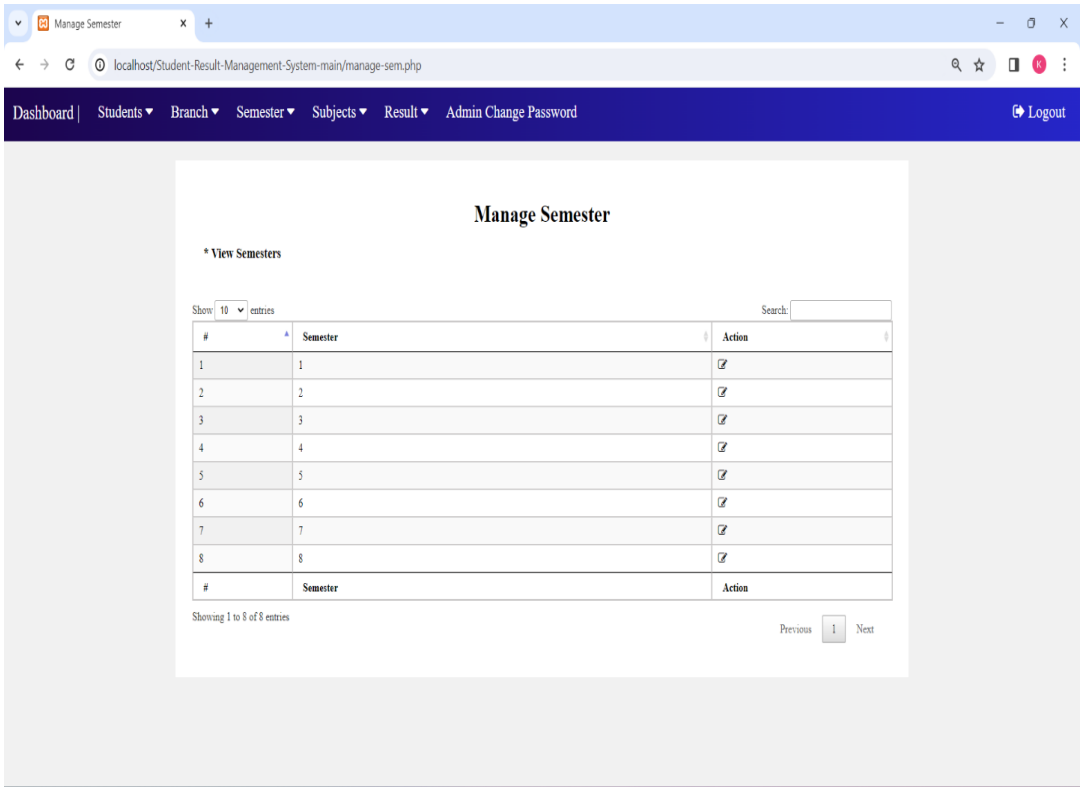
Edit Student's info: Selected Student record can be updated or deleted here



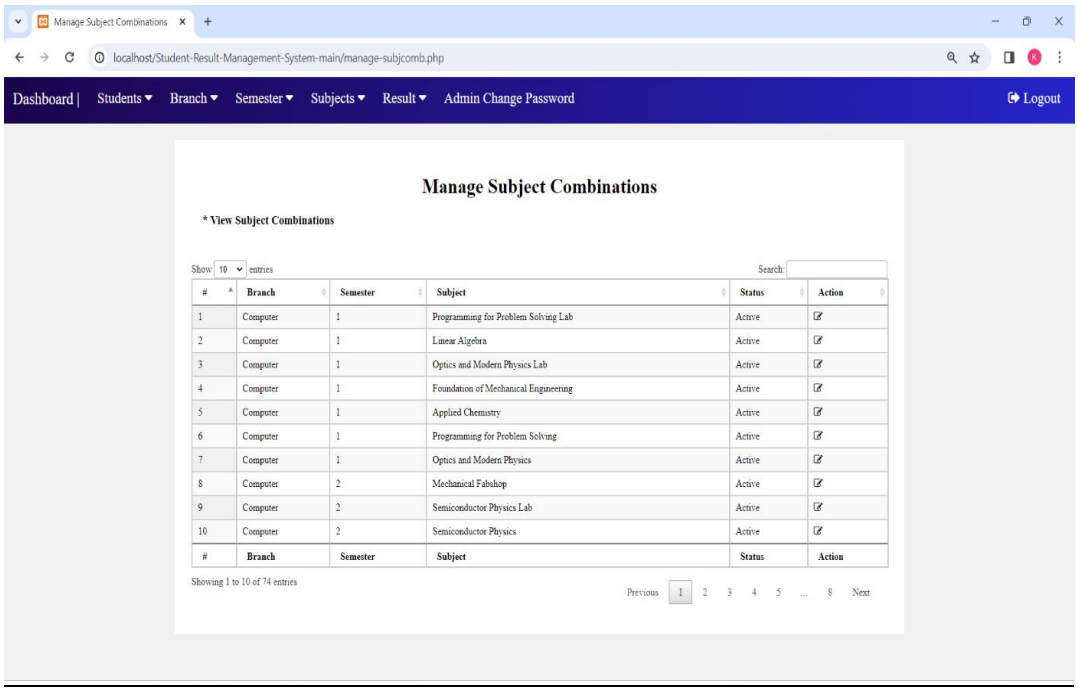
Manage Branch: Admin can view all the branches that exists



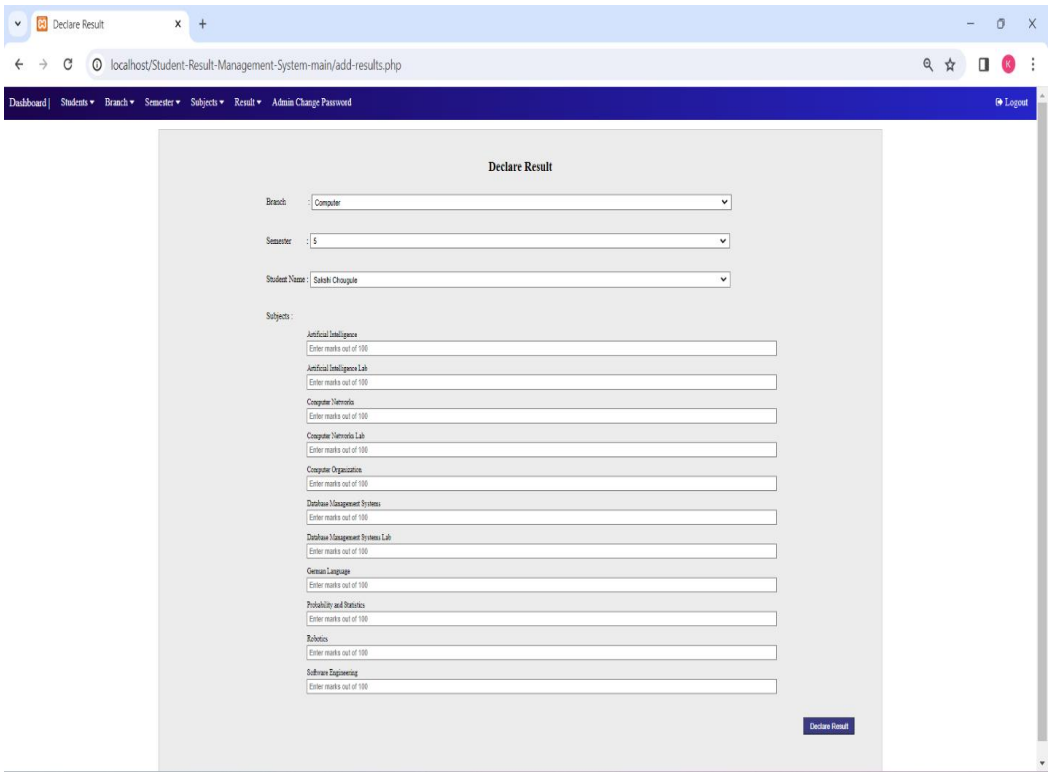
Manage Semester: All the added semesters can be viewed here



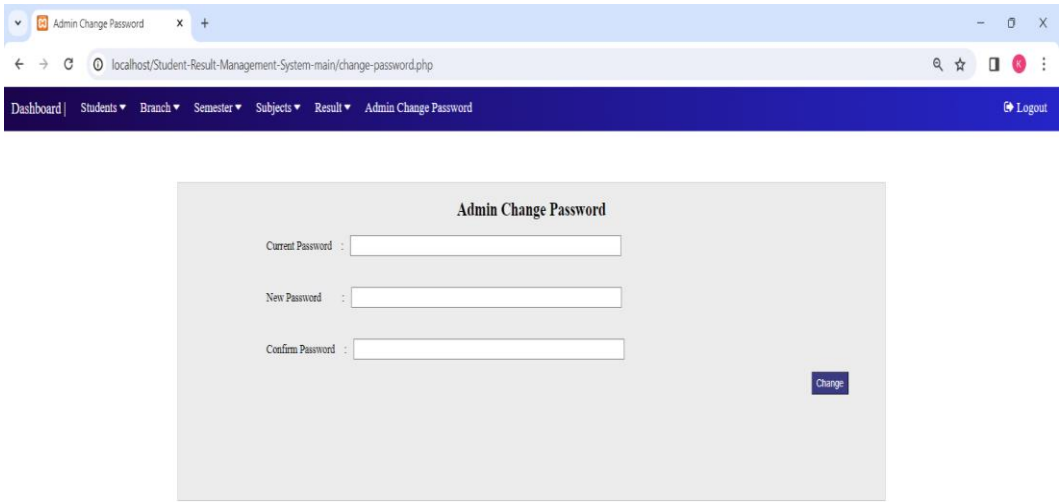
Manage Subject Combination: All the subject combinations for a particular branch and semester can be viewed here



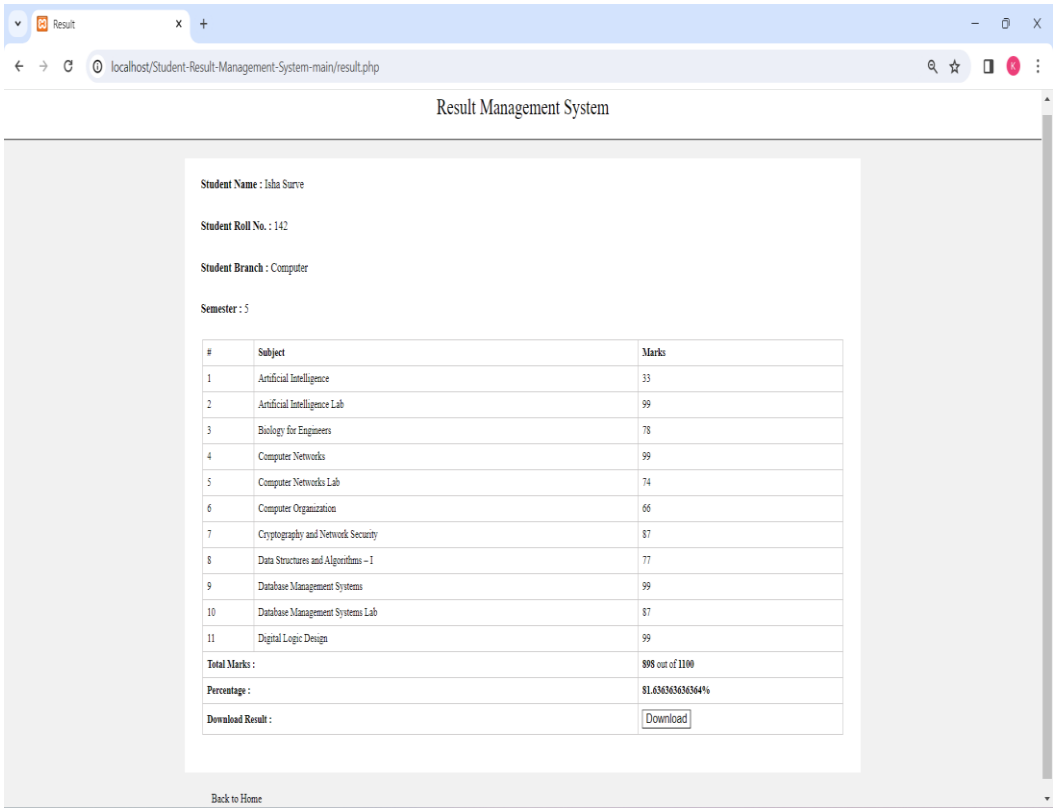
Add Result: Admin can declare result of any student in this page



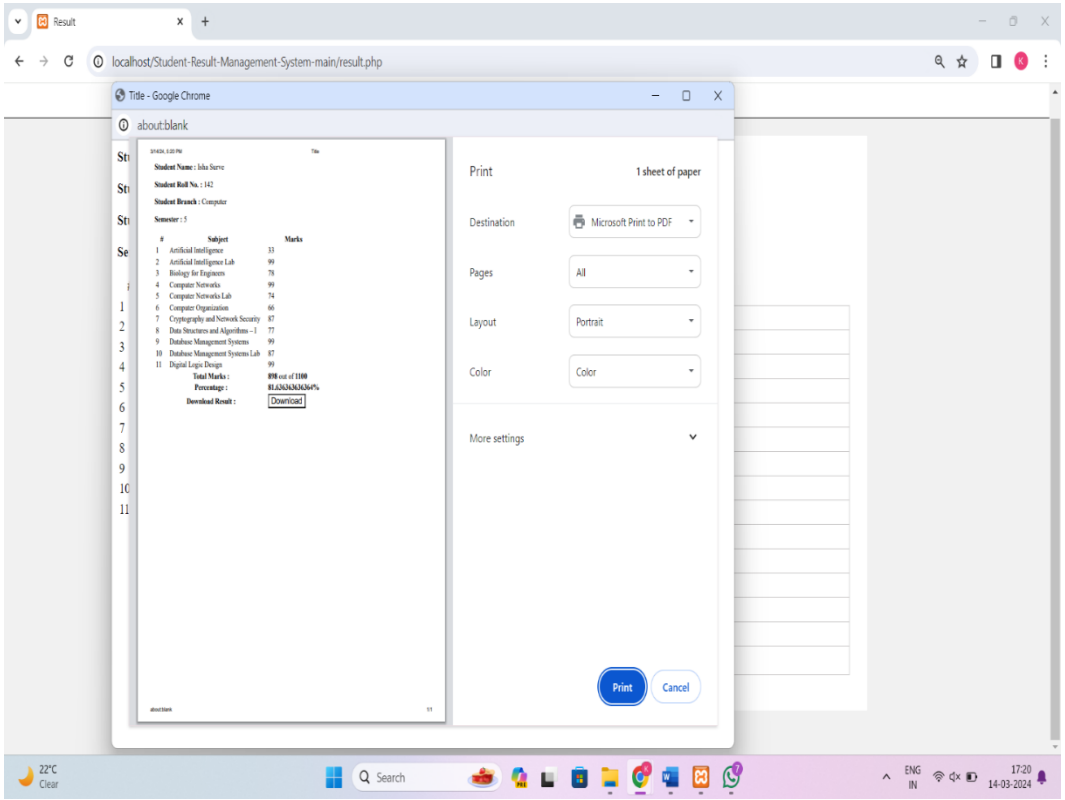
Admin Change Password: Admin is able to change his/her password in this page



Search Result: student's can search their results by providing necessary details



Print Result: Student result can also be printed/saved in pdf format



CONCLUSION

- The development of this system has effectively utilized practical knowledge to alleviate the efforts required by its users.
- Its user-friendly interface ensures that individuals familiar with web applications will find the system easy to navigate and utilize.
- The incremental modeling approach adopted for system development proves advantageous, allowing for easier modification and expansion of the current version. This facilitates the seamless deployment of enhanced versions following thorough testing and analysis of the existing system.
- The system's implementation has demonstrated a significant reduction in manual efforts, enhancing overall efficiency and productivity.
- Its intuitive design promotes accessibility for users of varying technical backgrounds, ensuring widespread adoption and satisfaction.

REFERENCE

1. Fundamentals of Database Systems by Rameez Elmasri and Shamkant Navathe 7th edition.
2. Database Management System 3rd Edition by Raghu Rama Krishna
3. https://www.w3schools.com/sql
4. www.studocu.com
5. https://www.w3schools.com/sql
6. https://chat.openai.com
7. https://app.diagrams.net