

# **Student Performance Management System**

**By**

**Mahedi Hasan Sabbir**

**ID: IT14021**

**Session:2014-15**

**Md. Forhad Hossain**

**ID: IT15012**

**Session:2014-15**

**Supervised by**

**Dr. Muhammad Shahin Uddin**

**Professor**

**Department of Information and Communication Technology**

**Mawlana Bhashani Science and Technology University**

**A project is submitted in partial fulfilment of the requirement for the degree of Bachelor of Science (Engg.) in Information and Communication Technology**



**Department of Information and Communication Technology(ICT)**

**Mawlana Bhashani Science and Technology University(MBSTU), Santosh, Tangail-1902,  
Bangladesh**

---

## **DECLARATION**

---

This is to certify that the project work entitled “Student Performance Management System” has been carried out by Mahedi Hasan Sabbir, IT-14021 and Md. Forhad Hossain, IT-15012 in the department of Information and Communication Technology (ICT), Mawlana Bhashani Science and Technology University (MBSTU), Santosh, Tangail-1902, Bangladesh.

The above research project work or any part of this work has not been submitted anywhere for the award of any degree or diploma.

### **Signature of Candidate**

Mahedi Hasan Sabbir

### **Signature of Candidate**

Md. Forhad Hossain

### **Signature of Supervisor**

Prof. Dr. Muhammad Shahin Uddin

Professor

Department of Information and Communication Technology (ICT)  
Mawlana Bhashani Science and Technology University (MBSTU), Santosh, Tangail-1902

---

## **APPROVAL**

---

This is to certify that the project work submitted by Mahedi Hasan Sabbir (IT-14021) and Md. Forhad Hossain (IT-15012) titled “Student Performance Management System” has been approved by the board of examiners for the partial fulfilment of the requirements for the degree of Bachelor of Science (Engineering) in the Department of Information and Communication Technology, Mawlana Bhashani Science and Technology University, Santosh, Tangail-1902, Bangladesh.

### **Examination Committee**

1. Dr. Muhammad Shahin Uddin  
Professor  
Department of ICT  
MBSTU, Santosh, Tangail-1902 ..... Chairman  
Examination Committee
  
2. Dr. Sajjad Waheed  
Professor  
Department of ICT  
MBSTU, Santosh, Tangail-1902 ..... Member(Internal)  
Examination Committee
  
3. Kawsar Ahmed  
Assistant Professor  
Department of ICT  
MBSTU, Santosh, Tangail-1902 ..... Member(Internal)  
Examination Committee
  
4. Dr. Fazlul Hasan Siddiqui  
Professor and Head  
Department of CSE  
Dhaka University of Engineering & Technology,  
Gazipur ..... Member(External)  
Examination Committee

---

## **ACKNOWLEDGEMENT**

---

All the praise and appreciation deserves to the almighty Allah for giving us patience in work, ability, and intelligence to complete as well as to submit my project work successfully without any major problem. We would like to express our sincere appreciation to our supervisor Dr. Muhammad Shahin Uddin, Professor, Department of Information and Communication Technology for giving us his valuable guidance and insight, simulation suggestion and advices, continuous encouragement and support and reliable throughout the thesis period to produce the report within a very short time. His guidance, punctuality, and simplicity have been indelible memory for us.

Finally, we would like to thank to all other honorable teachers, friends and those have helped, inspired and also given us mental support at different stages during completion of our project work.

Mahedi Hasan Sabbir

Md. Forhad Hossain

July, 2020

---

**ABSTRACT**

---

The student performance management and regulation of information about the educational process is an essential part of effective management of the educational institutions. It is the inevitable outcome of educational institutions reform to carry out the manual performance of the student has been carried on over the years, which are time consuming and vulnerable tradition of manual maintenance of student performance. To overcome the problems of manual performance, we have developed web based “Student Performance Management System”. The System is based on web server, which can be implemented on any computer. In this application, Tomcat is server side language, Oracle and Java is used as back-end design and HTML, CSS and JavaScript are used as front-end tools. The System can handle all the details about a student performance. The users will be provided with the separate username and password to make the students status. The System communicates with database residing on a remote server. The System facilitates the end users with interactive design and automated processing of performance management.

---

## PREFACE

This B.Sc. (Engg.) project is outlined based on Java and Oracle. This is carried out in the Department of Information and communication Technology at Mawlana Bhashani Science and Technology University in Santosh, Tangail-1902, Bangladesh.

This project includes nine chapters which are briefed as follows:

### **Chapter 1**

Chapter 1 provides a detailed discussion of Project Introduction.

### **Chapter 2**

Chapter 2 discusses about the System Analysis.

### **Chapter 3**

Chapter 3 provides a detailed discussions of Feasibility Study.

### **Chapter 4**

Chapter 4 discusses about the Tools and Technologies.

### **Chapter 5**

Chapter 5 discusses about Project Description.

### **Chapter 6**

Chapter 6 discusses about System Design (Front-End).

### **Chapter 7**

Chapter 7 discusses about System Design (Back-End).

### **Chapter 8**

Chapter 8 discusses about System Implementation.

### **Chapter 9**

Chapter 9 provides conclusion and future Enhancement to the discussion based on the projects.

---

## TABLE OF CONTENTS

---

Declaration.....	i
Approval.....	ii
Acknowledgement.....	iii
Abstract.....	iv
Preface.....	v
Table of Contents.....	vi-vii
List of Figures.....	viii-ix

### **Chapter 1: Introduction**

1.1 Problem and Motivation.....	1
1.2 Purpose and objectives .....	1
1.2.1 Purpose.....	1
1.2.2 Objective.....	1
1.3 Characteristics.....	1-2
1.4 Outline to Use This Book .....	2

### **Chapter 2: System Analysis**

2.1 Introduction.....	3
2.2 Existing System.....	3
2.3 Motivation of Manual System .....	4
2.3.1 Advantages of web based system.....	4

### **Chapter 3: Feasibility Study**

3.1 Introduction.....	5
3.1.2 Goals of Feasibility Study.....	5
3.2 Types of Feasibility Study.....	5
3.2.1 Economically Feasibility .....	6
3.2.2 Technical Feasibility .....	6
3.2.3 Legal Feasibility .....	6
3.2.4 Operational Feasibility .....	6-7
3.2.5 Scheduling Feasibility .....	7

### **Chapter 4: Tools and Technologies**

4.1 Java .....	8
4.1.1 IntelliJ IDEA.....	8
4.1.2 Oracle Database.....	9
4.1.3 Toad .....	9
4.2 The Sublime Text 3 Editor .....	9
4.3 HTML and CSS.....	10
4.4 JavaScript.....	10-11
4.5 Frameworks.....	11
4.5.1 Spring Boot.....	11-12
4.5.2 Bootstrap.....	12-13

4.5.3 Hibernate.....	13-14
<b>Chapter 5: Project Description</b>	
5.1 Problem Definition .....	15
5.2 Project Overview.....	15
5.3 Module Description.....	15
5.3.1 Teacher Module.....	15
5.3.2 Student Module.....	16
5.4 Data Flow Diagram.....	16-19
5.5 Input Design .....	20
5.6 Output Design .....	20
<b>Chapter 6: System Design (Front-End)</b>	
6.1 Registration Page.....	21
6.1.1 Teacher Registration Form.....	21
6.1.2 Student Registration Form .....	22
6.2 Login Page.....	22
6.3 Teacher Account Home Page.....	23
6.4 Teacher Account About Page.....	23
6.5 Teacher Course Page.....	24
6.5.1 Teacher Course Page After Taking Course.....	24
6.5.2 Teacher Create Class Page.....	25
6.5.3 Teacher's Class List Page.....	25
6.5.4 Teacher Create CT Page.....	26
6.5.5 Teacher Create Semester Final Page.....	26
6.6 Teacher contact Page .....	27
6.7 Student Account Home Page.....	28
6.8 Student About Page .....	28
6.9 Student Account Course Page .....	29
6.10 Student Performance Page.....	29
6.10.1 Student Performance Page.....	30
6.11 Student Contact Page.....	30
6.12 Search Page.....	31
6.12.1 Search Results Page.....	31
<b>Chapter 7: System Design (Back-End).....</b>	
.....32-37	
<b>Chapter 8: System Implementation</b>	
8.1 Purpose.....	38-39
8.2 System Maintenance.....	39
<b>Chapter 9: Conclusion and Future Work</b>	
9.1 Conclusion.....	40
9.2 Scope for Future Development.....	40
<b>Bibliography.....</b>	41

---

## **LIST OF FIGURES**

---

Figure No 5.4.1: Data Flow Diagram (Level 0).....	16
Figure No 5.4.2: Data Flow Diagram (Level 1).....	17
Figure No 5.4.3.1: Data Flow Diagram (Level 2).....	18
Figure No 5.4.3.2: Data Flow Diagram (Level 2).....	19
Figure No 6.1: Registration Page.....	21
Figure No 6.1.1 Teacher Registration Form.....	21
Figure No 6.1.2 Student Registration Form.....	22
Figure No 6.2 Login Page.....	22
Figure No 6.3 Teacher Account Home Page.....	23
Figure No 6.4 Teacher Account About Page .....	23
Figure No 6.5 Teacher Course Page.....	24
Figure No 6.5.1 Teacher Course Page After Taking Course.....	24
Figure No 6.5.2 Teacher Create Class Page.....	25
Figure No 6.5.3 Teacher's Class List Page.....	25
Figure No 6.5.4 Teacher Create CT Page.....	26
Figure No 6.5.5 Teacher Create Semester Final Page.....	26
Figure No 6.6 Teacher Contact Page.....	27
Figure No 6.7 Student Account Home Page .....	28
Figure No 6.8 Student About Page.....	28
Figure No 6.9 Student Account Course Page .....	29
Figure No 6.10 Student Performance Page.....	29

Figure No 6.10.1 Student Performance Page.....	30
Figure No 6.11 Student Contact Page.....	30
Figure No 6.12 Search Page.....	31
Figure No 6.12.1 Search Results Page.....	31
Figure No 7.1.1 Application Structure and Database Table.....	32
Figure No 7.1.2 Application Structure and Database Sequences.....	32
Figure No 7.1.3 Application Structure and Database Procedures.....	33
Figure No 7.2.1 Database Tables.....	33
Figure No 7.2.2 Database Packages.....	34
Figure No 7.2.3 Database Sequences.....	34
Figure No 7.2.4 Database Package Tree.....	35
Figure No 7.2.5 Database Package Heads.....	35
Figure No 7.2.6 Database Package Body.....	36
Figure No 7.2.7 Database Student Details Entity.....	36
Figure No 7.2.8 Database Teacher Details Entity.....	37
Figure No 7.2.9 Database Exam Details Entity.....	37

### **1.1 Problem and Motivation**

Management of student performance information is very important, because it is the cornerstone of good educational process. If this process is provided manually there is a big problem to handle the large number of students. It isn't just tedious, yet it is likewise untrustworthy and unsecured and furthermore, it can be lost.

To beat these issues, we have built up a superior framework which is Web-based, it is completely responsive where a client can use in versatile, tablets and diverse PC frameworks. In this framework, records have remained careful and secure without manually.

### **1.2 Purpose and Objectives**

#### **1.2.1 Purpose**

The student performance management system application will help in managing the student's reports, result and exam will become easier with one such system. The project aims to reduce the paper work and saving time to generate accurate result from the student's performance. All details about the student's and teachers will be available on a click both user and admin via internet. As the system is online the information is globally present to everyone. Every client will be given with a different username and password so it can be maintained properly.

#### **1.2.2 Objectives**

- The use of automated student performance management system in place of papers.
- Automatic calculation of performance.
- User friendly interface.
- Faster way to get performance.
- Reduce time work.
- To increase the security and make it reliable.

### **1.3 Characteristics**

The fundamental attributes of my created framework are that it is online, completely responsive and adaptable. It can be gotten to from any PC regardless of where you are. The day by day performance of student's calculated manually, which is time consuming and insecure. In Our project which is calculated automatically. Basically our project provide two modules such as user module(student) and admin module(teacher). A Student can only create his own profile. The

teachers can insert marks of his student and create their own profile. A student only can see on his own performance but teachers can see all students performance. Search option is another feature of our project which help us in emergency situation and it saves our time.

### **1.4 Outline to Use This Book**

If you are keen to get started, all you need is an enthusiasm to learn and a computer running Linux, Windows, Mac OS X, or Solaris. Each of these different operating systems supports Java and Oracle, as is explained later in the book.

In addition to this core platform, it is recommended that you take plenty of time to learn the different skills involved. Learning to bake in your head and solidify. Concepts that may seem obvious to some take a little longer to sink in with others, and you should allow yourself plenty of time to learn these different skills at your own pace.

Finally, it is recommended that you have a look around the Internet and join up on some of the Java/Oracle discussions forums and mailing lists. This will give you a great support mechanism when you don't understand certain concepts or need more help.

### **2.1 Introduction**

System Analysis is a described study of the various operations performed by a system and their relationships within and outside of the system. The investigation can be characterized as separating of any entire to discover their tendency, work and so on. It characterizes configuration as to make primer representations of; to portray an example of the layout for the arrangement. To plan and complete particularly by a creative course of action or in an apt divider. Framework investigation and configuration can be described as an arrangement of procedures and procedures, a network of premiums, a culture and a scholarly introduction. The different assignments in the framework examination incorporate the accompanying. Understanding Web based system.

- Planning.
- Scheduling.
- Developing candidate solution.
- Performing trade studies.
- Performing cost benefit analysis.
- Recommending alternative solutions.
- Selling of the system.
- Supervising, installing and maintaining the system.

This framework figures out how to keep the details about student and teacher and show it properly. First outline is the log in frame, home ,search and contact .This task will help the user to access the specific information. The Web-based framework will give adaptability for all user.

### **2.2 Existing System**

The Existing framework is a manual passage for showing the performance of the student. Here the students evaluation record will be done in the written by hand enrolls. It will be a monotonous activity to keep up the record for the teacher. The human exertion is more here. The recovery of the data isn't as simple as the records are kept up in the registers.

This web based system requires correct feed on input into the respective field. Suppose the wrong inputs are entered, the system resist to work. So, the user find it difficult to use.

### **2.3 Motivation of Manual System**

To defeat the disadvantages of the current framework, the proposed framework has been developed. This undertaking expects to lessen the printed material and sparing time to produce precise outcomes. The framework gives the best user interface. What's more, the proficient yield can be produced by utilizing this online framework.

#### **2.3.1 Advantages of Web based System**

- It is a relatively fast approach to get student performance.
- It is trouble free to use.
- It is highly reliable and secured system.
- Efficient reports.
- Search facilities.
- Best user Interface.

### **3.1 Introduction**

A feasibility study is an analysis that takes all of a project's relevant factors into account including economic, technical, legal, and scheduling considerations to ascertain the likelihood of completing the project successfully. It is simply an assessment of the practicality of a proposed plan or project. A possibility to examine is utilized to decide the feasibility of a thought, for example, guaranteeing a task is legitimately and actually practical and additionally monetarily reasonable. It lets us know whether a venture is justified regardless of the speculation-now and again, a task may not be feasible. There can be numerous purposes behind this, including requiring an excessive number of assets, which not just keeps those assets from performing different assignments yet, in addition, may cost in excess of an association would acquire back by going up against an undertaking that isn't painful.

Achievability examination starts once the objectives are characterized. It begins by creating expansive conceivable arrangements, which are conceivable to give a sign of what the new framework should resemble. This is the place innovativeness and creative ability are utilized. Examiners must concoct better approaches for doing things-produce new thoughts. There is no compelling reason to go into the itemized framework task yet. The arrangement ought to give enough data to make sensible gauges venture cost and give clients a sign of how the new framework will fit into the association. It is critical not to apply impressive exertion at this stage just to discover that the undertaking isn't beneficial or that there is a need altogether change the first objective.

#### **3.1.2 Goals of Feasibility Study**

The goals of feasibility studies are as follows:

- To understand thoroughly all aspects of a project, concept, or plan.
- To become aware of any potential problems that could occur while implementing the project.
- To determine if, after considering all significant factors, the project is viable that is, worth undertaking

#### **3.2 Types of Feasibility Study**

The practicality of another framework implies guaranteeing that the new framework, which we will execute, is proficient and moderate. There are different kinds of attainability to be resolved. They are,

### 3.2.1 Economically Feasibility

This appraisement typically involves a cost analysis of the project. Improvement of this Web-based framework is very economic practical. The main thing to be done is making a domain with a compelling supervision.

It is savvy as in has killed the printed material totally. The framework is additional time powerful in light of the fact that the estimations are mechanized which are made toward the finish of the month or according to the client prerequisite.

### 3.2.2 Technical Feasibility

This appraisement focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team is capable of converting the ideas into working systems. Technical feasibility also involves the evaluation of the hardware, software, and other technical requirements of the proposed system. The specialized prerequisite for the framework is monetary and it doesn't utilize some other extra Hardware and programming. Specialized assessment should likewise evaluate whether the current frameworks can be moved up to utilize the new innovation and whether the association has the mastery to utilize it.

Introduce all overhauls system into the .Net bundle upheld dowagers based application. This framework relies upon server and intranet benefit, database.

### 3.2.3 Legal Feasibility

This evaluation researches whether any part of the proposed venture clashes with lawful necessities like zoning laws, information insurance acts, or online networking laws. Suppose an association needs to develop another place of business in a particular area. A practicality study may uncover the association's optimal area isn't zoned for that kind of business.

That association has quite recently spared impressive time and exertion by discovering that their undertaking was not achievable ideal from the earliest starting point.

### 3.2.4 Operational Feasibility

The framework working is very simple to utilize and learn because of its basic yet appealing interface. The client requires no unique preparing for working the framework. Specialized execution incorporates issues, for example, deciding if the framework can give the correct data to the Department staff understudy subtle elements, and whether the framework can be sorted out

with the goal that it generally conveys this data at the ideal place and on time utilizing intranet administrations. Acknowledgment spins around the present framework and its staff.

#### **3.2.5 Scheduling Feasibility**

This appraisal is the most imperative for venture achievement; all things considered, a task will bomb if not finished on time. In planning practicality, an association appraises how much time the venture will take to finish.

At the point when these territories have all been analyzed, the attainability consider distinguishes any requirements the proposed venture may confront, including:

- Internal Project Constraints: Technical, Technology, Budget, Resource, and so forth.
- Internal Corporate Constraints: Financial, Marketing, Export, and so forth.
- External Constraints: Logistics, Environment, Laws and Regulations, and so forth.

### **4.1 Java**

Java is general-purpose programming language that is class-based, object-oriented, and designed to have as few implementation dependencies as possible. It is intended to let application developer write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java Virtual Machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but it has fewer low level facilities than either of them. Java is a popular programming language, created in 1995. It is owned by Oracle, and more than 3 billion devices run Java.

It is used for:

- Mobile applications (specially Android apps)
- Desktop applications
- Web applications
- Web servers and application servers
- Games
- Database connection
- And much, much more!

Why use Java?

- Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.)
- It is one of the most popular programming language in the world
- It is easy to learn and simple to use
- It is open-source and free
- It is secure, fast and powerful
- It has a huge community support (tens of millions of developers)
- Java is an object oriented language which gives a clear structure to programs and allows code to be reused, lowering development costs
- As Java is close to C++ and C#, it makes it easy for programmers to switch to Java or vice versa

#### **4.1.1 IntelliJ IDEA**

IntelliJ IDEA (hereafter referred as IntelliJ) is one of the most powerful and popular Integrated Development Environments (IDE) for Java. It was developed and is maintained by JetBrains, and is available in community and ultimate edition. This feature-rich IDE enables rapid development and helps in improving code quality. IntelliJ is a great JAVA IDEA that helps to developer better code Pros: Not only a great JAVA IDEA but also IntelliJ has support for other program languages like php, html, javascript. IntelliJ is easier to use and uses less memory making the user experience better.

### **4.1.2 Oracle Database**

Oracle Database is an RDBMS. An RDBMS that implements object-oriented features such as user-defined types, inheritance, and polymorphism is called an object-relational database management system (ORDBMS). Oracle Database is a multi-model database management system produced and marketed by Oracle Corporation. It is a database commonly used for running online transaction processing (OLTP), data warehousing (DW) and mixed (OLTP & DW) database workloads. The latest generation, Oracle Database 19c, is available on-prem, on-cloud, or in a hybrid-Cloud environment. 19c may also be deployed on Oracle Engineered Systems on-prem, on Oracle (public) cloud or (private) cloud at customer.

### **4.1.3 Toad**

TOAD is a set of development tools used for application development, database development, or business intelligence and assist developers in deploying Oracle-based applications and Web services on the Windows platform. It is a database management toolset from Quest Software that database developers, database administrators, and data analysts use to manage both relational and non-relational databases using SQL. There are Toad products for developers and DBAs, which run on Oracle, SQL server, IBM DB2 (LUW & z/OS), SAP and MySQL, as well as, a Toad product for data preparation, which supports most data platforms. Toad solutions enable data professionals to automate processes, minimize risks, and cut project delivery time.

## **4.2 The Sublime Text 3 editor**

Sublime Text is a cross-stage source code proofreader written in C++ and Python. It initially underpins a lot of programming and markup dialects, and its usefulness can be expanded by means of clients with modules.

Sublime Text 3 has two fundamental highlights that are sheet administration and image ordering.

Through sheet administration clients are to move between sheets by hotkeys and image ordering empower Sublime Text to check records and construct a file to make simple the highlights Go to Symbol and Go to Definition in venture.

It is downloaded from [www.sublimetext.com/3](http://www.sublimetext.com/3), webpage. A rundown of a few highlights of Sublime Text is as following,

- It is cross stage (Linux, Mac OS X, Windows)
- Compatible with various dialect designers shape Text Mate
- Project particular inclinations
- And others.

### **4.3 HTML and CSS**

HTML remains for Hypertext Markup Language and CSS remains for Cascading Style Sheets are the urgent innovations for making site pages. HTML supplies the structure of the page, and CSS the format, for a decent variety of gadgets. Together with scripting and illustrations, HTML and CSS are the essences of building Web Applications and Web pages. HTML gives creators and engineers the accompanying offices,

- To configuration frames for coordinating exchanges with remote administrations, for use in reserving the spot, scanning for data, requesting items, and others.
- Retrieving the web data through hypertext joins.
- To incorporate video and sound clasps, spreadsheets, and different applications straight into their archives.
- The designer can distribute online archives with content, headings, tables, photographs, and others.

CSS portrays the Web pages introduction, including design, hues, and textual styles. It empowers the fashioner to change the introduction to different kinds of gadgets, similar to a little screen, expansive screens, or printers.

CSS is separate from HTML, and their partition makes it simple to protect and look after destinations, share templates crosswise over pages, and suit pages to different situations.

### **4.4 JavaScript**

JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform.

Javascript is must for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning Javascript:

- Javascript is the most popular programming language in the world and that makes it a programmer's great choice. Once you learnt Javascript, it helps you developing great front-end as well as back-end softwares using different Javascript based frameworks like jQuery, Node.JS etc.

- Javascript is everywhere, it comes installed on every modern web browser and so to learn Javascript you really do not need any special environment setup. For example Chrome, Mozilla Firefox , Safari and every browser you know as of today, supports Javascript.
- Javascript helps you create really beautiful and crazy fast websites. You can develop your website with a console like look and feel and give your users the best Graphical User Experience.
- JavaScript usage has now extended to mobile app development, desktop app development, and game development. This opens many opportunities for you as Javascript Programmer.
- Due to high demand, there is tons of job growth and high pay for those who know JavaScript. You can navigate over to different job sites to see what having JavaScript skills looks like in the job market.
- Great thing about Javascript is that you will find tons of frameworks and Libraries already developed which can be used directly in your software development to reduce your time to market.

### 4.5 Frameworks

#### 4.5.1 Spring Boot

Spring Boot is a lightweight framework that takes most of the work out of configuring Spring-based applications. It provides a good platform for Java developers to develop a stand-alone and production-grade spring application that you can just run. You can get started with minimum configurations without the need for an entire Spring configuration setup.

It offers the following amenities to its developers-

- Easy to understand and enlarge spring applications
- Enhancement productivity
- Reduces the development time

It provides a flexible way to configure Java Beans, XML configurations, and Database Transactions. In Spring Boot, everything is auto configured; no manual configurations are needed. It includes Embedded Servlet Container.

It automatically configures our application based on the dependencies we have added to the project by using **@EnableAutoConfiguration** annotation.

Observe the following code for a better understanding –

```
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.EnableAutoConfiguration;

@EnableAutoConfiguration
public class StudentPerformanceApplication {
    public static void main(String[] args) {
        SpringApplication.run(StudentPerformanceApplication, args);
    }
}
```

}

The entry point of the spring boot application is the class contains `@SpringBootApplication` annotation and the main method.

Observe the following code for a better understanding –

```
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class StudentPerformanceApplication {
    public static void main(String[] args) {
        SpringApplication.run(StudentPerformanceApplication, args);
    }
}
```

Spring Boot automatically scans all the components included in the project by using `@ComponentScan` annotation.

Observe the following code for a better understanding –

```
import org.springframework.boot.SpringApplication;
import org.springframework.context.annotation.ComponentScan;

@ComponentScan
public class StudentPerformanceApplication {
    public static void main(String[] args) {
        SpringApplication.run(StudentPerformanceApplication, args);
    }
}
```

### **4.5.2 Bootstrap**

Bootstrap is front-end structure and gathering of instruments and systems for building web applications. It comprises of HTML and CSS based plan formats for routes, shapes, catches, typography, and other interface components, and furthermore JavaScript expansions.

Bootstrap is a free and open-source front-end system (library) for outlining sites and web applications. It contains HTML-and CSS-based outline formats for typography, frames, catches, route and other interface segments, and in addition discretionary JavaScript expansions. Not at all like many web systems, it worries about front-end advancement as it were. Bootstrap is the second-most-featured venture on GitHub, with in excess of 123,000 stars.

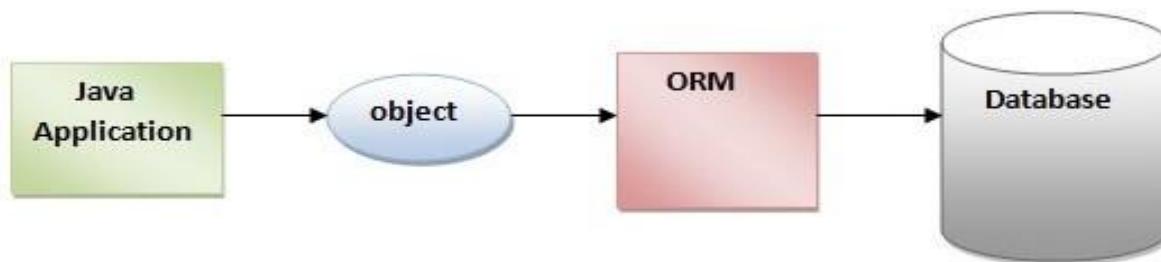
Bootstrap was produced by Mark Otto and Jacob Thornton and named Twitter Blueprint. Prior to the Bootstrap structure, creators were utilizing diverse libraries for interface improvement, which had numerous irregularities and their upkeep was troublesome.

On 31 Jan 2012, Bootstrap 2 was discharged. This system has conveyed numerous progressions to the current parts and, likewise, included 12 segment matrix format and responsive plan constituents. On August 19, 2013, Bootstrap 3 was declared, which moved to the principal approach of portable and utilizing a level outline. The principal alpha variant of Bootstrap was spread out on 19, Aug 2015. The main beta rendition was discharged on 10 August 2017. The steady form of Bootstrap 4 was discharged on January 18, 2018, fundamentally enhancing print styles, fringe utilities and gives more control over the flexbox.

### 4.5.3 Hibernate

Hibernate is a Java framework that simplifies the development of Java application to interact with the database. Hibernate is a high-performance Object/Relational persistence and query service, which is licensed under the open source GNU Lesser General Public License (LGPL) and is free to download. Hibernate not only takes care of the mapping from Java classes to database tables (and from Java data types to SQL data types), but also provides data query and retrieval facilities. It is an open source, lightweight, ORM (Object Relational Mapping) tool. Hibernate implements the specifications of JPA (Java Persistence API) for data persistence.

An ORM tool simplifies the data creation, data manipulation and data access. It is a programming technique that maps the object to the data stored in the database.



Hibernate supports almost all the major Relational Database Management System(RDBMS). Following is a list of few of the database engines supported by Hibernate –

- HSQL Database Engine
- DB2/NT
- MySQL
- PostgreSQL
- FrontBase
- Oracle
- Microsoft SQL Server Database
- Sybase SQL Server
- Informix Dynamic Server

Hibernate supports a variety of other technologies, including –

- XDoclet Spring
- J2EE
- Eclipse plug-ins
- Maven

### 5.1 Problem Definition

This framework created will lessen the manual work and maintain a strategic distance from excess information. By keeping up the participation physically, at that point, effective reports can't be produced. The framework can create proficient week by week, solidify report in view of the participation.

### 5.2 Project Overview

Student's Performance Management System basically has two main modules for proper functioning

- Teacher module
- Student module

### 5.3 Module Description

The framework ought to be composed such that lone approved individuals ought to be permitted to get to some specific modules. The records ought to be changed by just overseers and nobody else. The client ought to dependably be responsible for the application and not the other way around. The UI ought to be predictable so the client can deal with the application effortlessly and speed. The application ought to be outwardly, reasonably clear.

#### 5.3.1 Teacher Module

- In this module teacher can take any course by giving department, session, semester and course code from select option.
- In this module teacher can't take course which is taken by he/she or another teacher.
- In this module teacher can view his/her taken course.
- In this module teacher can create class, in create class surface shows all student who take this particular course and teacher can tick mark who attend class after this he/she can save it.
- In this module teacher can create CT1, CT2, CT3, CT4 and Semester Final exam and can marking all student.
- In teacher module he/she can view all student's performance.
- In this module teacher can view all teacher and student's information.

**5.3.2 Student Module**

- In this module teacher can take any course by giving session, semester and course code from select option.
- In this module student can't take course which is taken by he/she.
- In this module student can view his courses.
- In this module student can view his/her performance of any particular course.
- In this module student can view all teacher and student's information except students performance.

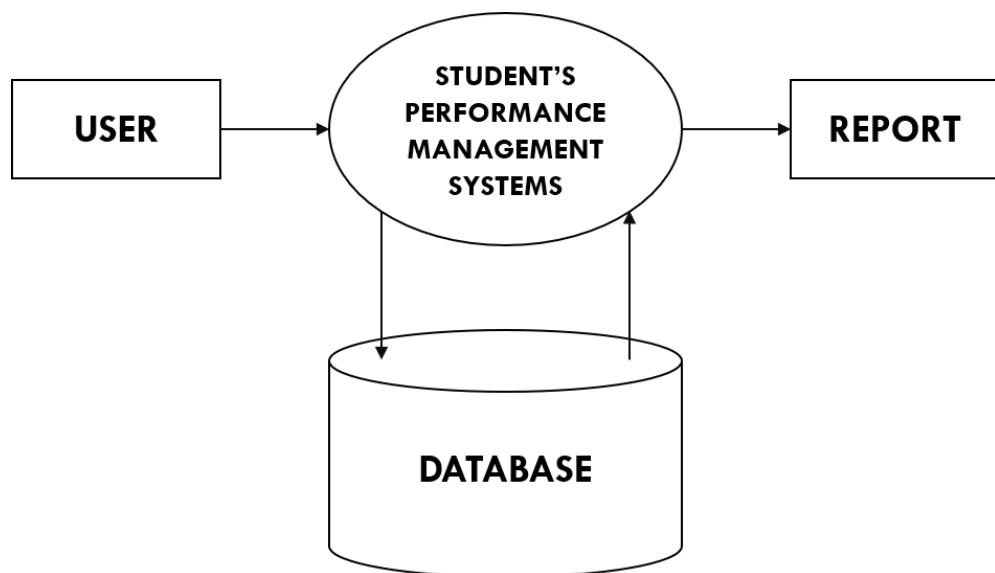
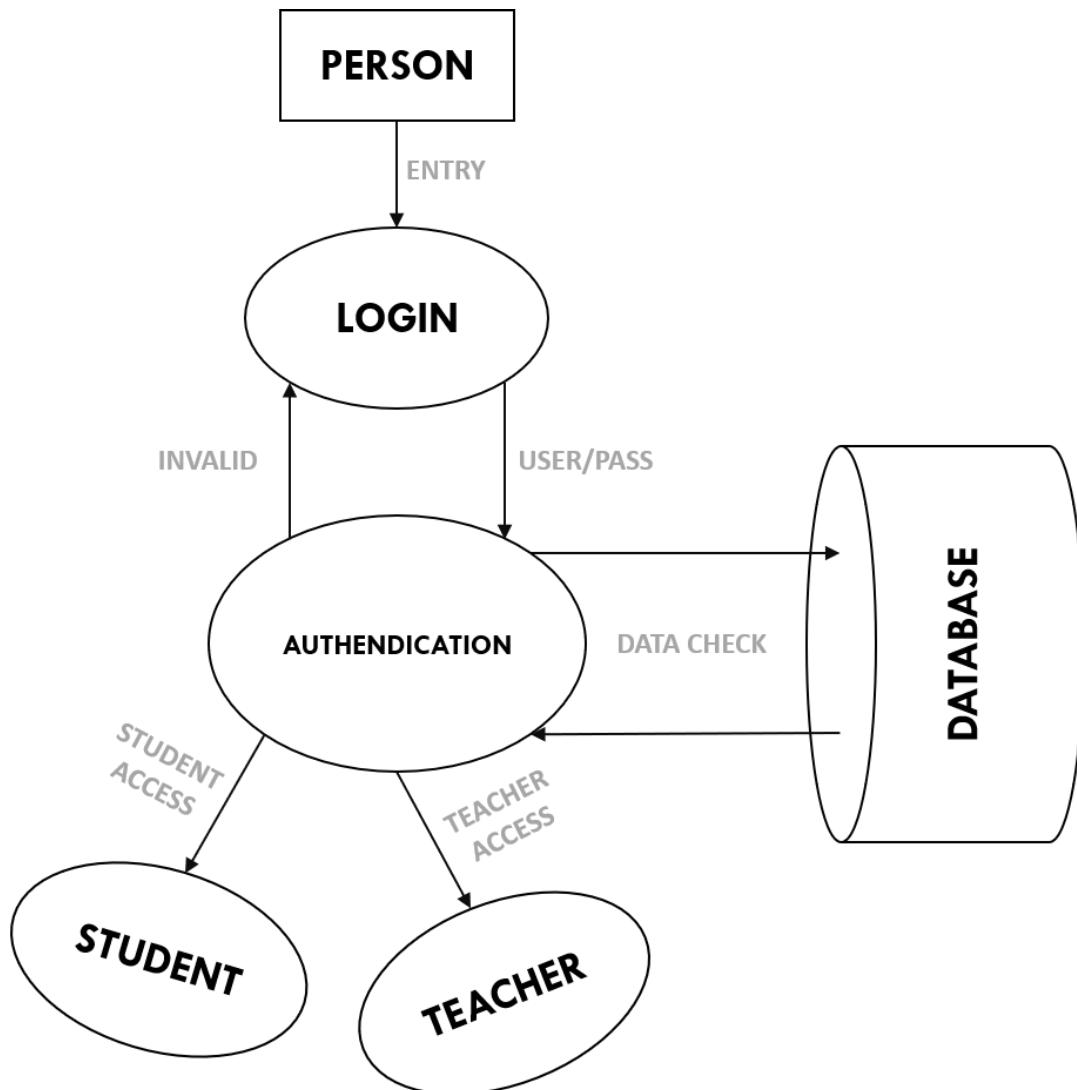
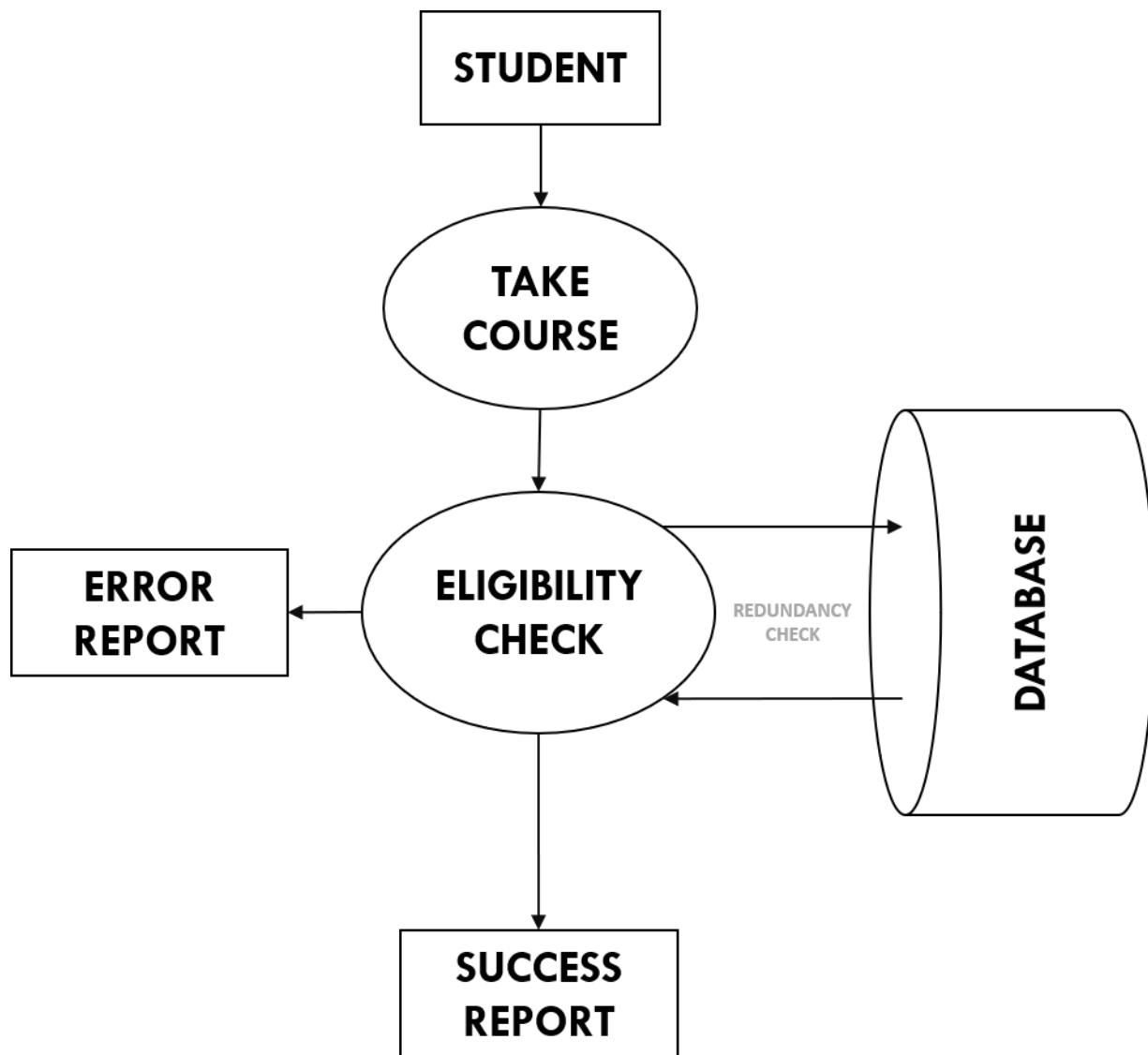
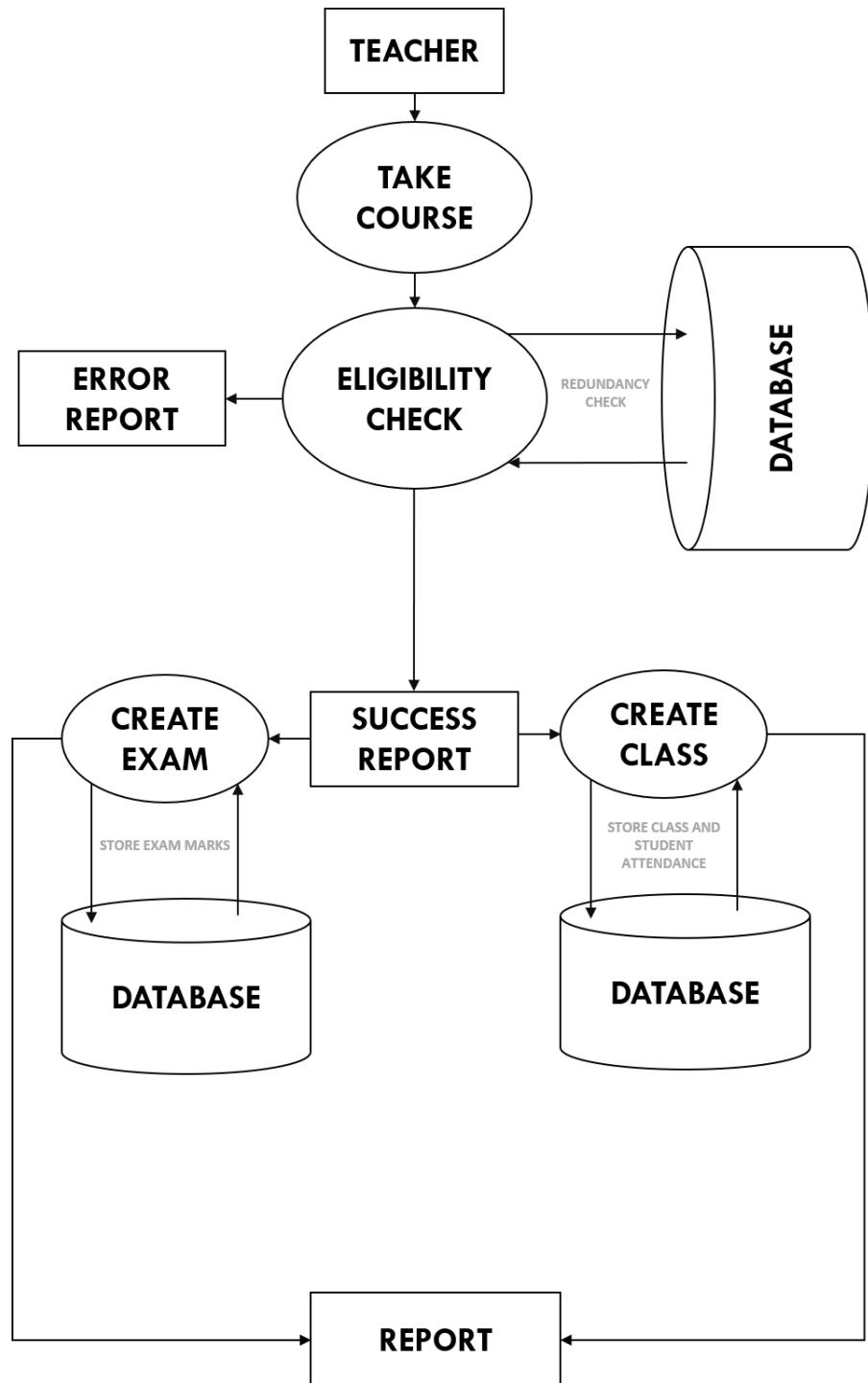
**5.4 Data Flow Diagram****5.4.1 DFD Level 0:**

Figure 5.4.1-Data Flow Diagram Level 0

**5.4.2 DFD Level 1:***Figure 5.4.2-Data Flow Diagram Level 1*

**5.4.3.1 DFD Level 2:***Figure 5.4.3.1-Data Flow Diagram Level 2*

**5.4.3.2 DFD Level 2:***Figure 5.4.3.2 - Data Flow Diagram Level 2*

### 5.5 Input Design

Information configuration is a piece of in general framework outline that requires extraordinary consideration planning input information is to make the information entered simple and free from mistakes. The info shapes are outlined utilizing the controls accessible in PHP and JavaScript structure. Approval is made for every single information that is entered. Help data is accommodated the clients amid when the client feels troublesome.

Info configuration is the way toward changing over the client began contributions to a PC based organization. A framework client interfacing through a workstation must have the capacity to advise the framework whether to acknowledge the contribution to deliver reports. The accumulation of information is thought to be the most costly piece of the framework plan. Since the information must be arranged in such a way to get applicable data, extraordinary care is taken to get relevant data.

This venture first will enter to the contribution of designation frames it will be made on under study points of interest shape and subject section shape, timetable frame .it will compute subject insightful participation framework. Next one if you need any check on your information's additionally accessible in subtle elements demonstrate frames. Participation in entered single subject savvy or all subject insightful participation framework accessible in this undertaking.

### 5.6 Output Design

Output design this web based “**Student Performance Management System**” generally refers to the results and information that are generated by the system for many end-users; output is the main reason for developing the system and the basis on which they evaluate the usefulness of the application.

The output is designed in such a way that it is attractive, convenient and informative. Forms are designed with various features, which make the console output more pleasing.

As the outputs are the most important sources of information to the users, better design should improve the system’s relationships with us and also will help in decision making. Form design elaborates the way output is presented and the layout available for capturing information.

One of the most important factors of the system is the output it produces. This system refers to the results and information generated. Basically the output from a computer system is used to communicate the result of processing to the user.

Attendance management system to show the report subject wise attendance maintaining by staffs. Taken all in all report get on ahead benefits as it were. This structures will demonstrate week by week report and merge report produced the date, clump, and class shrewd to our end client. We need to change our answer to change over Excel design on the off chance that you need change any alteration.

### 6.1 Registration Page

Both teacher and student can registration from here.

The screenshot shows a registration form titled "Sign Up". At the top right, there is a "Log In" button and a user icon. The form fields include:

- Enter name
- Enter email
- Enter mobile
- DOB dd----yyyy
- blood group
- Registration Type

At the bottom of the form, a success message reads "Successfully registered as a Teacher.". Below the form, a footer bar states "Developed By Mahedi Hasan Sabbir & Md Forhad Hossain | Student's Performance".

Figure 6.1- Registration Page

#### 6.1.1 Teacher Registration Form

The screenshot shows a detailed teacher registration form titled "Sign Up". At the top right, there is a "Log In" button and a user icon. The form fields include:

- Dr. Muhammad Shahin Uddin
- shahin.mbstu@gmail.com
- 01759852318
- DOB 03-May-1977
- O (+ve)
- Teacher
- Engineering
- ICT
- Join Date 04-May-2007
- Professor
- shahin\_uddin
- .....
- .....

At the bottom right of the form is a large orange "Sign Up" button.

Figure 6.1.1- Teacher Registration Form

### 6.1.2 Student Registration Form

The screenshot shows a registration form titled "Sign Up" for "Student's Performance". The form fields include:

- Name: MD. FORHAD HOSSAIN
- Email: forhadhossain2845@gmail.com
- Mobile: 01558951885
- Date of Birth (DOB): 31-Dec-1997
- Blood Group: O (+ve)
- Category: Student
- Stream: Engineering
- Subject: ICT
- Class: 12TH
- Session: 2014-15
- Join Date: 22-Feb-2015
- Department: IT
- Roll Number: 15012
- Username: forhad\_hossain
- Password: ..... (twice)

A "Sign Up" button is located at the bottom right.

Figure 6.1.2- Student Registration Form

### 6.2 Login Page

Both teacher and student can login here.

The screenshot shows a login form titled "Sign In" for "Student's Performance". The form fields include:

- User Type: Teacher
- Username: shahin\_uddin
- Password: .....

A "Sign In" button is located at the bottom right. Below the form, a link says "Don't have an account? Sign Up".

At the bottom of the page, a teal footer bar contains the text: "Developed By Mahedi Hasan Sabbir & Md Forhad Hossain | Student's Performance".

Figure 6.2- Login Page

### 6.3 Teacher Account Home Page

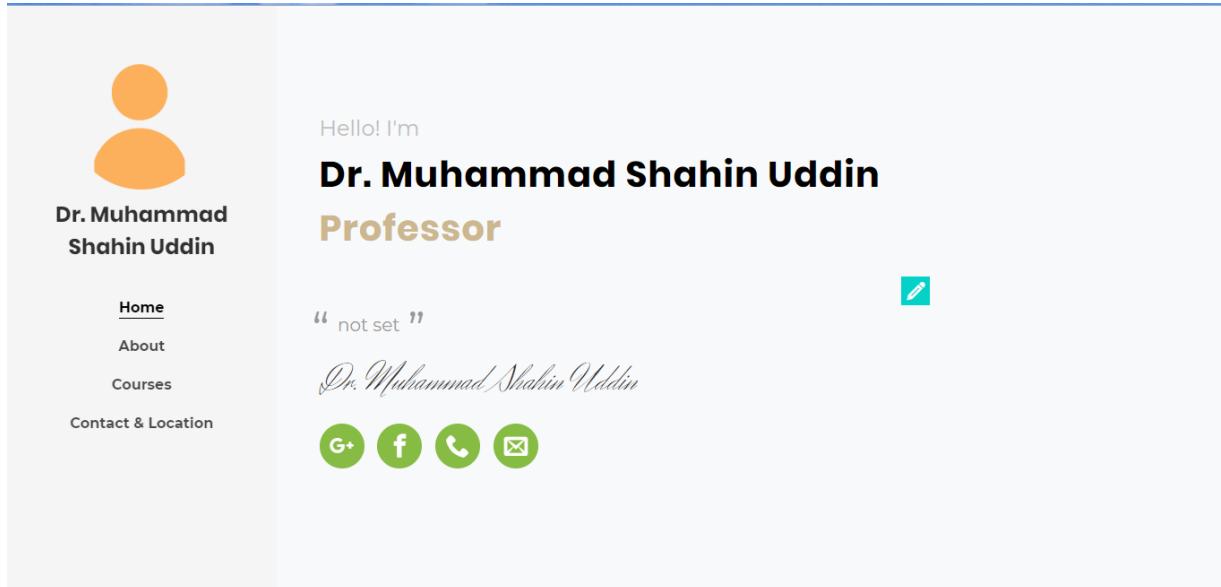


Figure 6.3- Teacher Account Home Page

### 6.4 Teacher Account About Page

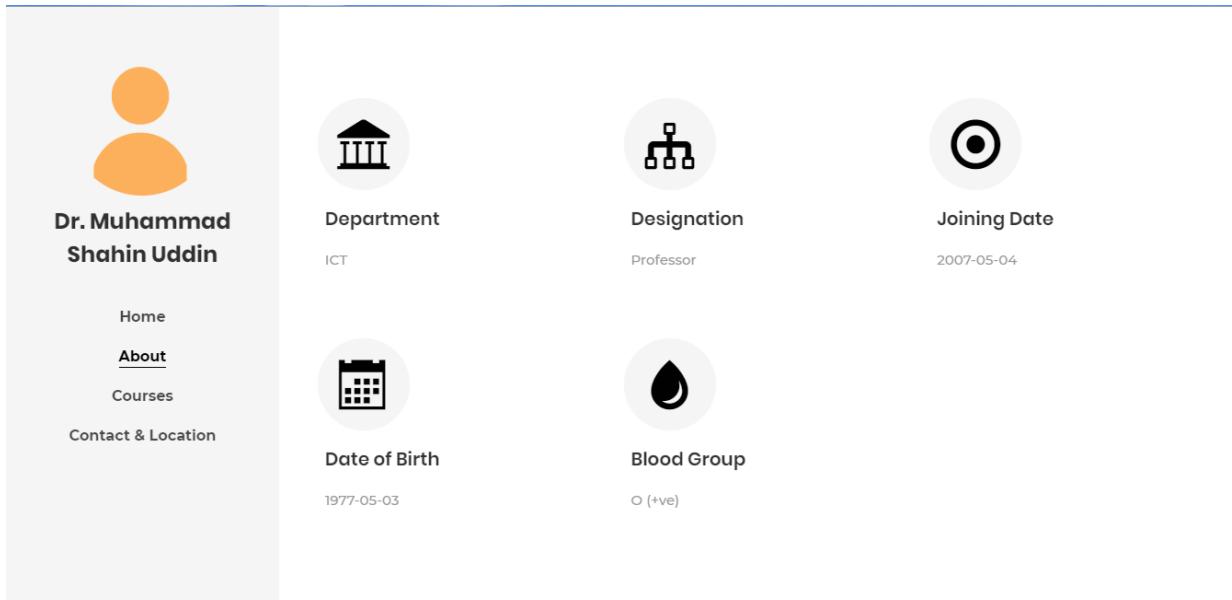


Figure 6.4- Teacher Account About Page

### 6.5 Teacher Course Page

Teacher can take course from here.

The screenshot shows a user profile for 'Dr. Muhammad Shahin Uddin' on the left, featuring a yellow circular icon. The main content area is titled 'My Courses' with a subtitle 'HOME COURSES'. It includes buttons for 'Add New Course' and 'Search Your Course'. A table lists courses with columns for Session, Course Code, Course Title, View, and Delete. The 'Courses' link in the sidebar is underlined.

Session	Course Code	Course Title	View	Delete
2015-16	ICT-1101	Basic Electrical Circuit		
2015-16	ICT-1102	Basic Electrical Circuit Lab		
2014-15	ICT-2101	Operational Amplifier and Timer Circuit		
2014-15	ICT-2102	Operational Amplifier and Timer Circuit Lab		

Figure 6.5- Teacher Course Page

#### 6.5.1 Teacher Course Page After Taking Course

The screenshot shows the same user profile and sidebar as Figure 6.5. The main content area now displays a message 'Successfully Taken' above the course list. The course table includes an additional column for 'View' (represented by eye icons) and 'Delete' (represented by X icons). The 'Courses' link in the sidebar is underlined.

Session	Course Code	Course Title	View	Delete
2015-16	ICT-1101	Basic Electrical Circuit		
2015-16	ICT-1102	Basic Electrical Circuit Lab		
2014-15	ICT-2101	Operational Amplifier and Timer Circuit		
2014-15	ICT-2102	Operational Amplifier and Timer Circuit Lab		

Figure 6.5.1- Teacher Course Page After Taking Course

### 6.5.2 Teacher Create Class Page

Teacher can create class from here with student's attendance.

The screenshot shows a user interface for managing student attendance. On the left, there is a sidebar with a profile picture and the name "Dr. Muhammad Shahin Uddin". Below the name are links for "Home", "About", "Courses" (which is underlined), and "Contact & Location". The main content area has a title "My Courses" at the top. Below it is a table with columns: "Serial", "Student Id", "Name", and "Attendance". The table contains six rows of data:

Serial	Student Id	Name	Attendance
1	IT-14021	MAHEDI HASAN SAI	<input checked="" type="checkbox"/>
2	IT-15005	Sarwar Jahan	<input type="checkbox"/>
3	IT-15012	MD. FORHAD HOSS.	<input checked="" type="checkbox"/>
4	IT-15021	Md Piyal	<input checked="" type="checkbox"/>
5	IT-15035	Tuhin Islam	<input type="checkbox"/>
6	IT-15036	Arif Nurullah	<input checked="" type="checkbox"/>

At the top of the table, there are "Save" and "Cancel" buttons.

Figure 6.5.2- Teacher Create Class Page

### 6.5.3 Teacher's Class List Page

The screenshot shows a browser window displaying the "My Courses" page. The URL in the address bar is "localhost:8085/student/performance/teacher/account/md\_shahin/courses". The page layout is similar to the previous one, with a sidebar on the left and a main content area on the right. The main content area has a title "My Courses" and a navigation bar with "HOME" and "COURSES". Below the navigation bar is a "Create New Class" button. At the top of the main content area, there are tabs for "CT - 1", "CT - 2", "CT - 3", "CT - 4", and "Semester Final". The "Semester Final" tab is highlighted. The main content area contains a table with columns: "Class No", "Date", "Edit", and "View". The table has seven rows of data:

Class No	Date	Edit	View
1	07/07/2020		
2	07/07/2020		
3	07/07/2020		
4	07/07/2020		
5	07/07/2020		
6	07/07/2020		
7	07/07/2020		

Figure 6.5.3-Teacher's Class List Page

#### 6.5.4 Teacher Create CT Page

Teacher can create CT from here with student's marks.

The screenshot shows a user interface for managing student marks. On the left, there is a sidebar with a profile icon and the text "Dr. Mohammad Shahin Uddin". Below the profile are links: Home, About, Courses, and Contact & Location. The main content area has a header "HOME COURSES" and a title "My Courses". Below the title is a table with two rows of data. The table has columns for Serial, Student Id, Name, and Marks. The first row has a Serial of 1, Student Id of IT-14021, Name of Mahedi Hasan Sabbir, and a Marks field containing the value 15. The second row has a Serial of 2, Student Id of IT-15013, Name of Md Forhad Hossain, and a Marks field containing the value 16. At the top of the table are "Save" and "Cancel" buttons.

Serial	Student Id	Name	Marks
1	IT-14021	Mahedi Hasan Sabbir	15
2	IT-15013	Md Forhad Hossain	16

Figure 6.5.4- Teacher Create CT Page

#### 6.5.5 Teacher Create Semester Final Page

Teacher can create Semester Final from here with student's marks.

The screenshot shows a browser window titled "Dr. Mohammad Shahin Uddin" with the URL "localhost:8085/student/performance/teacher/account/md\_shahin/courses". The page layout is identical to Figure 6.5.4, featuring a sidebar with a profile icon and the text "Dr. Mohammad Shahin Uddin", and a main content area with a "My Courses" section. The table below the title shows two rows of student marks. The first row has a Serial of 1, Student Id of IT-14021, Name of Mahedi Hasan Sabbir, and a Marks field containing the value 47. The second row has a Serial of 2, Student Id of IT-15013, Name of Md Forhad Hossain, and a Marks field containing the value 46. The "Save" and "Cancel" buttons are at the top of the table.

Serial	Student Id	Name	Marks
1	IT-14021	Mahedi Hasan Sabbir	47
2	IT-15013	Md Forhad Hossain	46

Figure 6.5.5- Teacher Create Semester Final Page

### 6.6 Teacher Contact Page

Views teacher's contact information here and teacher can edit them from this page.

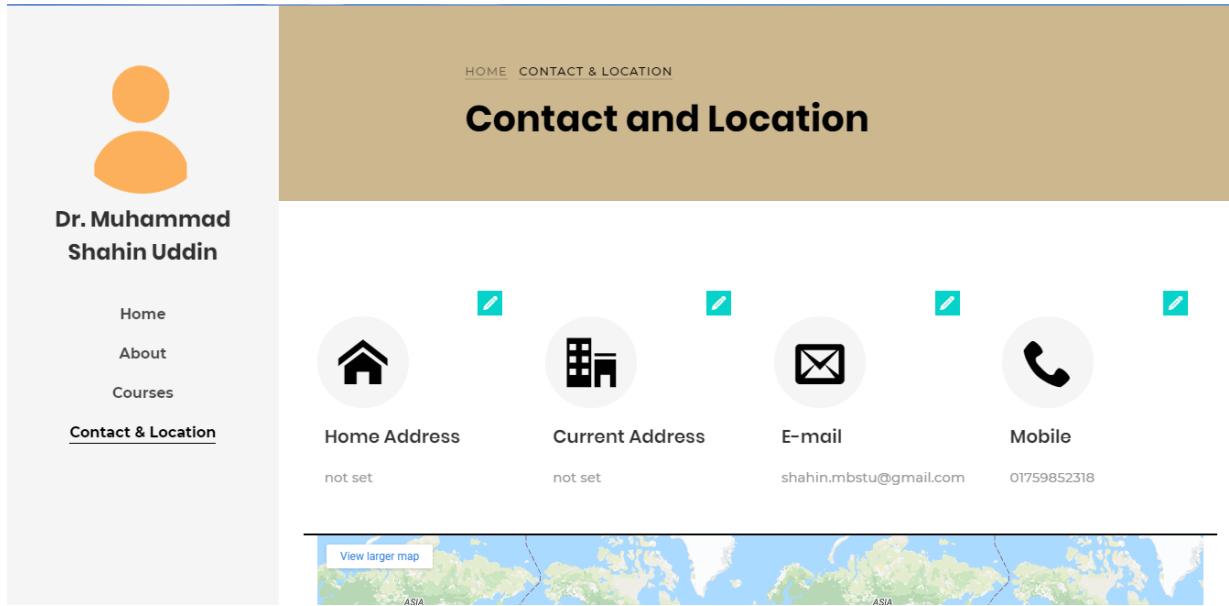


Figure 6.6-Teacher Contact Page

### 6.7 Student Account Home Page

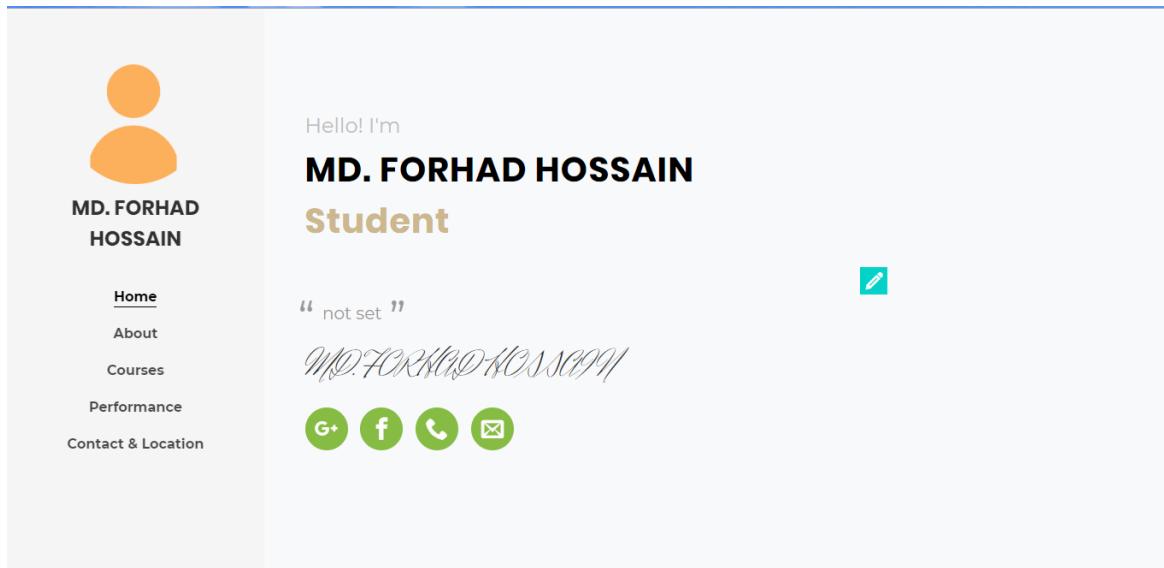


Figure 6.7- Student Account Home Page

### 6.8 Student About Page

View student's basic information from here.

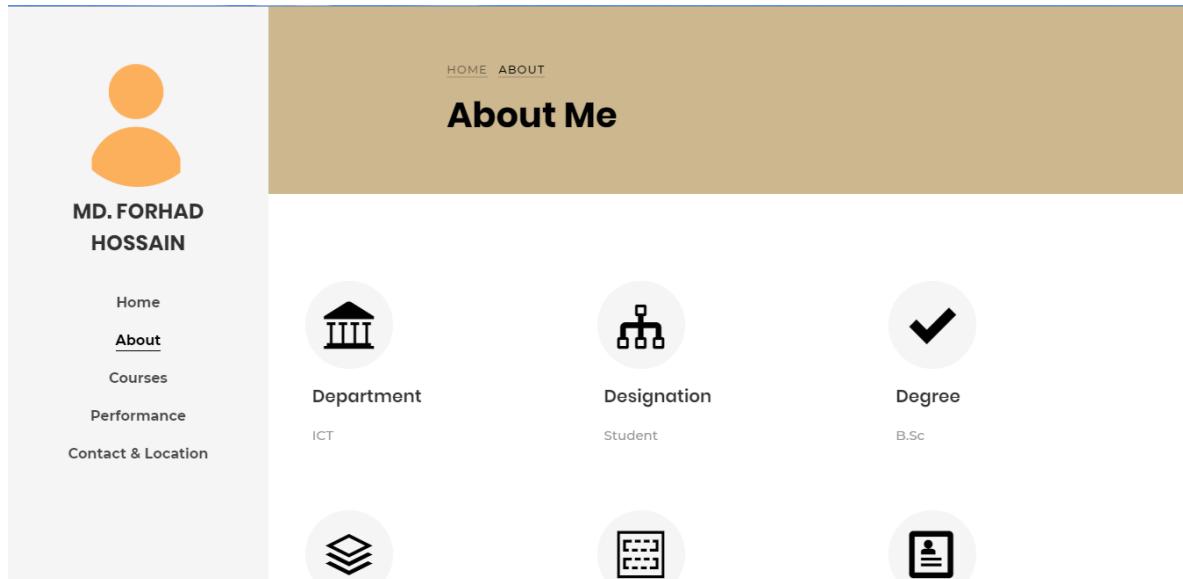


Figure 6.8- Student About Page

### 6.9 Student Account Course Page

Student can take course from this page which are not taken by he/she.

The screenshot shows a user profile for 'MD. FORHAD HOSSAIN' on the left, featuring a placeholder profile picture and a menu with links to Home, About, Courses (which is underlined), Performance, and Contact & Location. The main content area is titled 'My Courses' and displays a table of successfully taken courses. The table has columns for Session, Course Code, Course Title, View, and Delete. Two entries are listed:

Session	Course Code	Course Title	View	Delete
2014-15	ICT-2101	Operational Amplifier and Timer Circuit		
2014-15	ICT-2102	Operational Amplifier and Timer Circuit Lab		

Figure 6.9- Student Account Course Page

### 6.10 Student's Performance Page

Student can view his/her performance from this page.

The screenshot shows a web browser window with the URL 'localhost:8085/student/performance/student/profile/mh\_sabbir/performance'. The title bar says 'Basic Electrical Circuit'. On the left, there is a sidebar with four buttons: Attendance (green), Class Test (red), Semester Final (blue), and Average (light blue). To the right is a large donut chart with three segments. The innermost segment is dark blue and labeled 'Attendance 86%'. The middle segment is red, and the outermost segment is light blue. The background of the main content area is white.

Figure 6.10- Student's Performance Page

### 6.10.1 Student's Performance Page

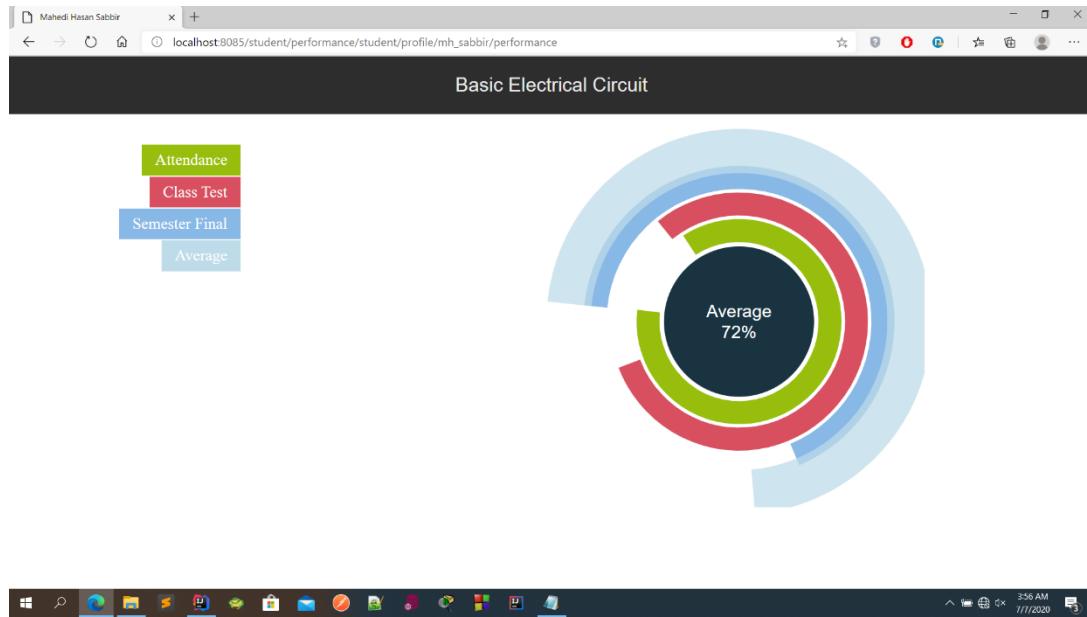


Figure 6.10.1- Student's Performance Page

### 6.11 Student Contact Page

Shows student's contact information here and student can edit them from this page.

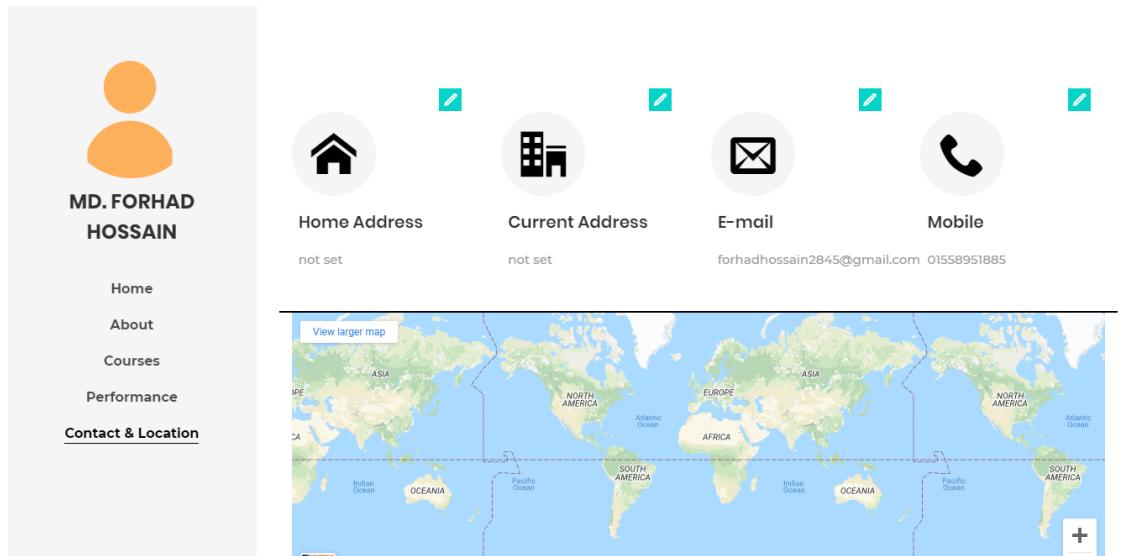


Figure 6.11- Student Contact Page

### 6.12 Search Page

Teacher/ Student can search teacher/student from here according to their department, semester, name, student id, blood group.

The screenshot shows a dark purple header with the text "Search Profile". Below it is a white search form. The form includes a placeholder "search here", two dropdown menus ("Student" and "Student ID"), an input field containing "IT-14021", and a blue "SEARCH" button. At the bottom, a teal footer bar displays the text "Developed By Mahedi Hasan Sabbir & Md Forhad Hossain | Student's Performance".

Figure 6.12- Search Page

#### 6.12.1 Search Results Page

Shows search result in this page.

The screenshot shows a purple header with navigation links: "Home", "Search", "Contact", "Logout", and a user icon. Below the header is a user profile section featuring a large orange circular icon. To the right of the icon, the text "MAHEDI HASAN SABBIR" is displayed, followed by "12TH, ICT" in smaller font. The background of the main content area is light gray.

Figure 6.12.1- Search Results Page

### 7.1 Application

#### 7.1.1 Application Structure and Database Table

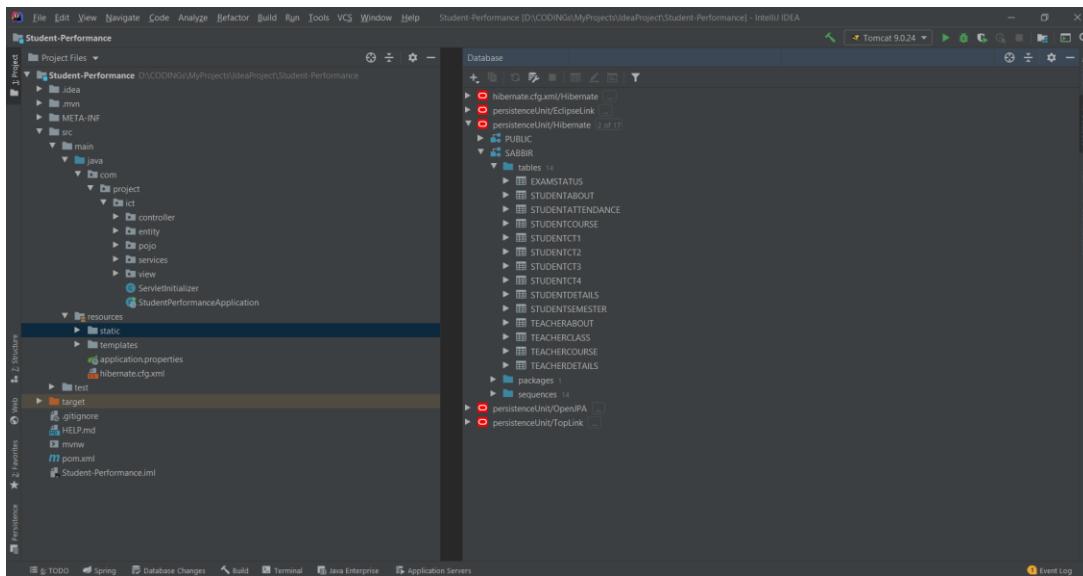


Figure 7.1.1 Application Structure and Database Table

#### 7.1.2 Application Structure and Database Sequences

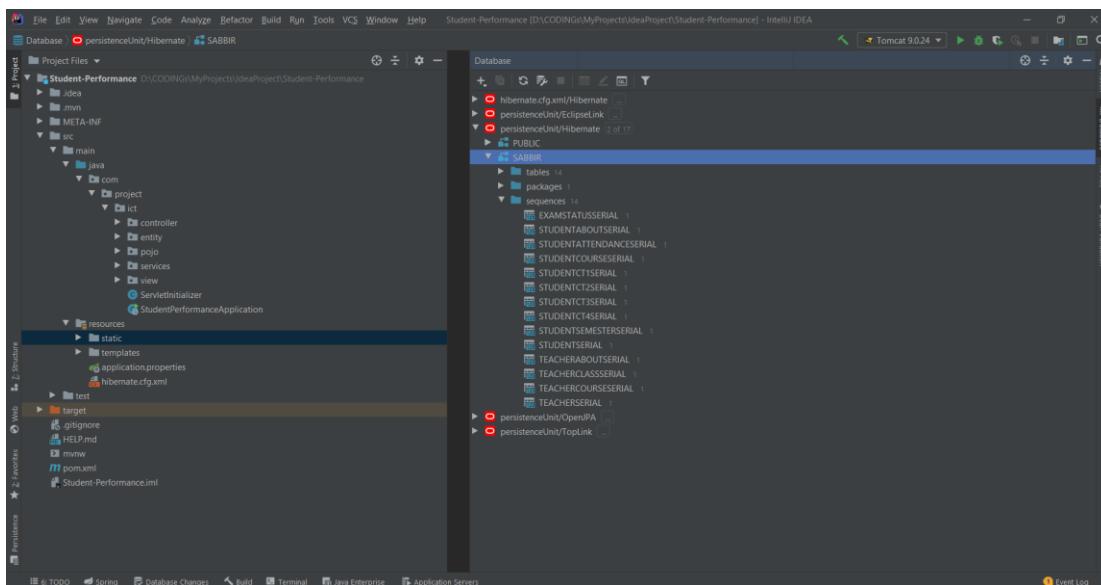


Figure 7.1.2- Application Structure and Database Sequences

### 7.1.3 Application Structure and Database Procedures

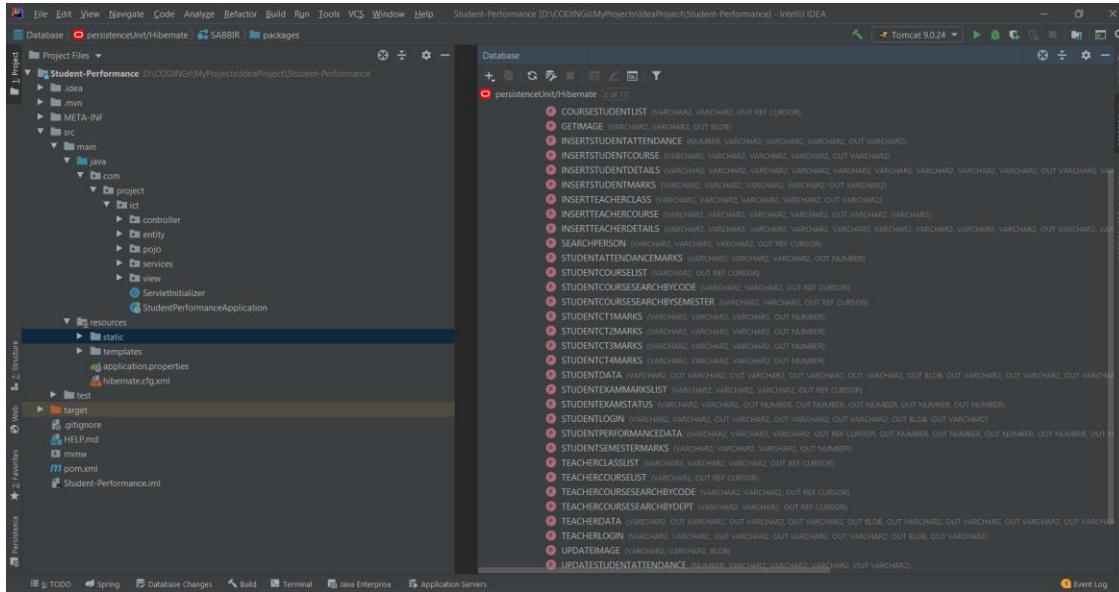


Figure 7.1.3- Application Structure and Database Procedures

## 7.2 Database

### 7.2.1 Database Tables

Table	Schema	Tablespace	Last Analyzed	Num Rows	Created	Last DDL
EXAMSTATUS	SABBIR	SYSTEM	7/7/2020 10:00:39 PM	11	7/5/2020 8:40:26 PM	7/5/2020 8:40:26 PM
STUDENTABOUT	SABBIR	SYSTEM	7/7/2020 10:00:35 PM	4	7/5/2020 8:40:23 PM	7/5/2020 8:40:23 PM
STUDENTATTENDANCE	SABBIR	SYSTEM	7/7/2020 10:00:37 PM	3	7/5/2020 8:40:23 PM	7/5/2020 8:40:23 PM
STUDENTCOURSE	SABBIR	SYSTEM	7/7/2020 10:00:40 PM	6	7/5/2020 8:40:23 PM	7/5/2020 8:40:23 PM
STUDENTCT1	SABBIR	SYSTEM	7/7/2020 10:00:38 PM	3	7/5/2020 8:40:23 PM	7/5/2020 8:40:23 PM
STUDENTCT2	SABBIR	SYSTEM	7/7/2020 10:00:38 PM	3	7/5/2020 8:40:23 PM	7/5/2020 8:40:23 PM
STUDENTCT3	SABBIR	SYSTEM	7/6/2020 10:00:41 PM	1	7/5/2020 8:40:24 PM	7/5/2020 8:40:24 PM
STUDENTCT4	SABBIR	SYSTEM	7/5/2020 10:42:32 PM	5	7/5/2020 8:40:24 PM	7/5/2020 8:40:24 PM
STUDENTDETAILS	SABBIR	SYSTEM	7/7/2020 10:00:36 PM	4	7/5/2020 8:40:22 PM	7/5/2020 8:40:22 PM
STUDENTSSEMESTER	SABBIR	SYSTEM	7/7/2020 10:00:39 PM	3	7/5/2020 8:40:25 PM	7/5/2020 8:40:25 PM
TEACHERABOUT	SABBIR	SYSTEM	7/7/2020 10:31:25 PM	2	7/5/2020 8:40:26 PM	7/5/2020 8:40:26 PM
TEACHERCLASS	SABBIR	SYSTEM	7/7/2020 10:00:40 PM	11	7/5/2020 8:40:26 PM	7/5/2020 8:40:26 PM
TEACHERCOURSE	SABBIR	SYSTEM	7/7/2020 10:31:26 PM	2	7/5/2020 8:40:26 PM	7/5/2020 8:40:26 PM
TEACHERDETAILS	SABBIR	SYSTEM	7/7/2020 10:31:25 PM	2	7/5/2020 8:40:25 PM	7/5/2020 8:40:25 PM

Figure 7.2.1- Database Tables

## 7.2.2 Database Packages

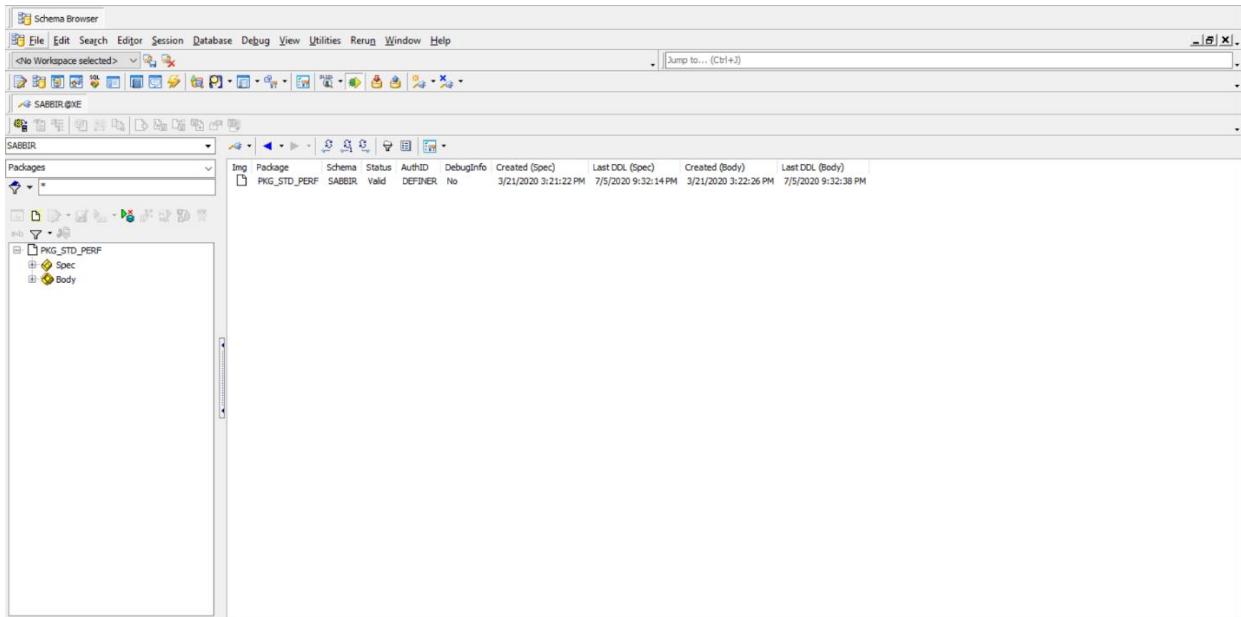


Figure 7.2.2- Database Packages

## 7.2.3 Database Sequences

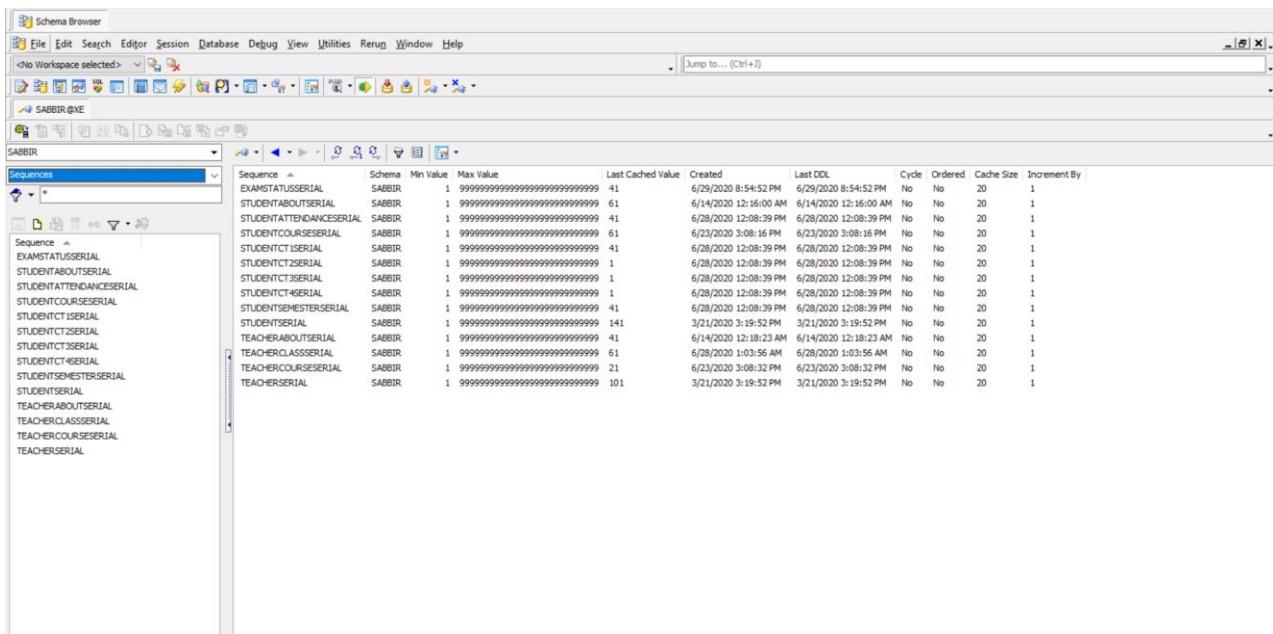


Figure 7.2.3- Database Sequences

### 7.2.4 Database Package Tree

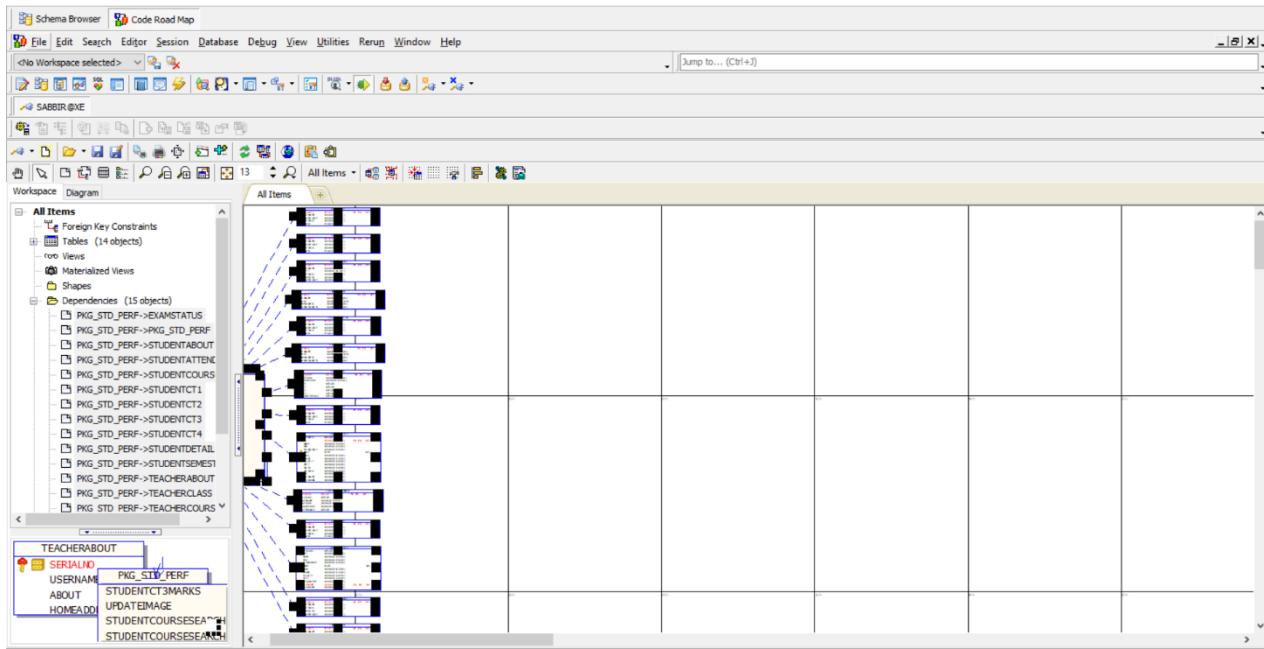


Figure 7.2.4- Database Package Tree

### 7.2.5 Database Package Heads

The screenshot shows the Oracle SQL Developer interface. The left pane is titled 'Packages' and displays a tree view of package heads, specifically 'PKG\_STD\_PERF'. The right pane shows the detailed code for this package. The code includes several procedures:

```

145 PROCEDURE updateStudentAttendance (inClassNo      IN      INT,
146                                         inUsername    IN      VARCHAR2,
147                                         inSession    IN      VARCHAR2,
148                                         inCourseCode IN      VARCHAR2,
149                                         pResult      OUT     VARCHAR2);
150
151
152 PROCEDURE insertStudentMarks (inSaveType     IN      VARCHAR2,
153                               inUsername    IN      VARCHAR2,
154                               inSession    IN      VARCHAR2,
155                               inCourseCode IN      VARCHAR2,
156                               pResult      OUT     VARCHAR2);
157
158 PROCEDURE updateStudentMarks (inSaveType     IN      VARCHAR2,
159                               inUsername    IN      VARCHAR2,
160                               inSession    IN      VARCHAR2,
161                               inCourseCode IN      VARCHAR2,
162                               inMarks      IN      INT,
163                               pResult      OUT     VARCHAR2);
164
165 PROCEDURE studentPerformanceData (inUsername   IN      VARCHAR2,
166                                    inSession    IN      VARCHAR2,
167                                    inCourseCode IN      VARCHAR2,
168                                    outStatusCursor OUT    SYS_REFCURSOR,

```

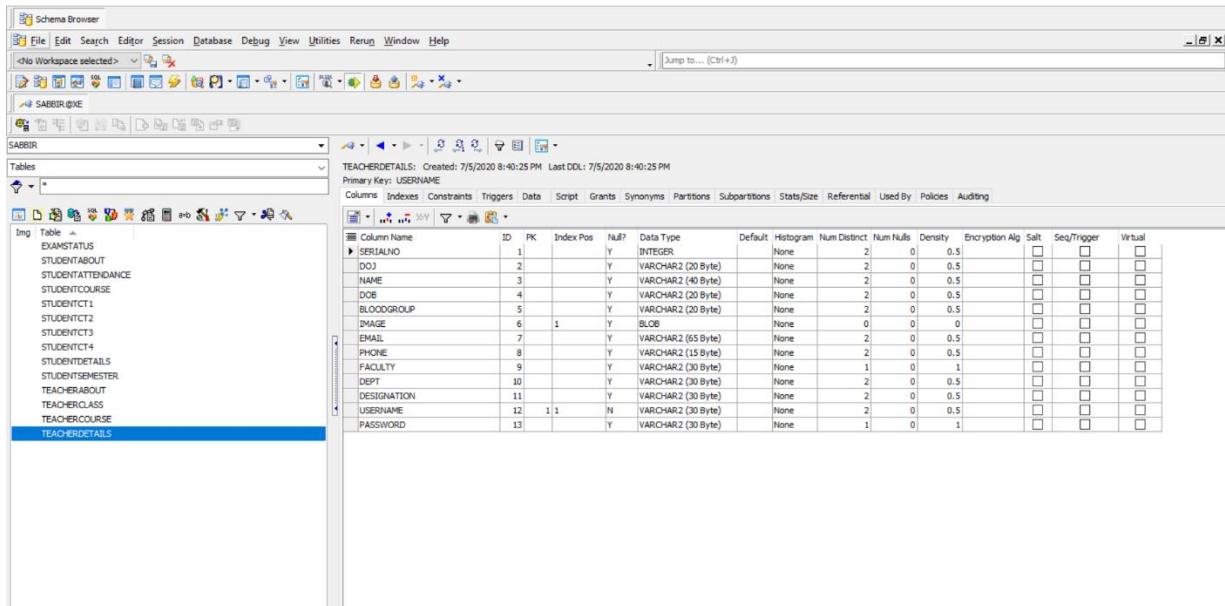
Figure 7.2.5- Database Package Heads

## 7.2.6 Database Package Body

```

21      thisSid StudentDetails.sid%TYPE;
22      BEGIN
23          SELECT sid
24              INTO thisSid
25              FROM StudentDetails
26             WHERE sid = inSid;
27
28
29      IF thisSid = inSid
30      THEN
31          pResult := 'duplicateSid';
32          RETURN;
33      END IF;
34  EXCEPTION
35      WHEN NO_DATA_FOUND
36      THEN
37          DECLARE
38              CURSOR usernameCheck
39                  IS
40                      SELECT username FROM StudentDetails;
41          BEGIN
42              FOR i IN usernameCheck
43                  LOOP
44                      IF i.username = inUsername
45                  THEN
46                      pResult := 'usernameExists';
47                      RETURN;
48                  END IF;
49              END LOOP;
50          END;
51      END IF;
52  END;
53
54  IF pResult = 'usernameExists'
55  THEN
56      RETURN;
57  END IF;
58
59  -- Additional code for cursor handling follows
60
61  END;
62
63  IF pResult = 'usernameExists'
64  THEN
65      RETURN;
66  END IF;
67
68  -- Additional code for cursor handling follows
69
70  END;
71
72  IF pResult = 'usernameExists'
73  THEN
74      RETURN;
75  END IF;
76
77  -- Additional code for cursor handling follows
78
79  END;
80
81  IF pResult = 'usernameExists'
82  THEN
83      RETURN;
84  END IF;
85
86  -- Additional code for cursor handling follows
87
88  END;
89
90  IF pResult = 'usernameExists'
91  THEN
92      RETURN;
93  END IF;
94
95  -- Additional code for cursor handling follows
96
97  END;
98
99  IF pResult = 'usernameExists'
100 THEN
101    RETURN;
102 END IF;
103
104 -- Additional code for cursor handling follows
105
106 END;
107
108 IF pResult = 'usernameExists'
109 THEN
110    RETURN;
111 END IF;
112
113 -- Additional code for cursor handling follows
114
115 END;
116
117 IF pResult = 'usernameExists'
118 THEN
119    RETURN;
120 END IF;
121
122 -- Additional code for cursor handling follows
123
124 END;
125
126 IF pResult = 'usernameExists'
127 THEN
128    RETURN;
129 END IF;
130
131 -- Additional code for cursor handling follows
132
133 END;
134
135 IF pResult = 'usernameExists'
136 THEN
137    RETURN;
138 END IF;
139
140 -- Additional code for cursor handling follows
141
142 END;
143
144 IF pResult = 'usernameExists'
145 THEN
146    RETURN;
147 END IF;
148
149 -- Additional code for cursor handling follows
150
151 END;
152
153 IF pResult = 'usernameExists'
154 THEN
155    RETURN;
156 END IF;
157
158 -- Additional code for cursor handling follows
159
160 END;
161
162 IF pResult = 'usernameExists'
163 THEN
164    RETURN;
165 END IF;
166
167 -- Additional code for cursor handling follows
168
169 END;
170
171 IF pResult = 'usernameExists'
172 THEN
173    RETURN;
174 END IF;
175
176 -- Additional code for cursor handling follows
177
178 END;
179
180 IF pResult = 'usernameExists'
181 THEN
182    RETURN;
183 END IF;
184
185 -- Additional code for cursor handling follows
186
187 END;
188
189 IF pResult = 'usernameExists'
190 THEN
191    RETURN;
192 END IF;
193
194 -- Additional code for cursor handling follows
195
196 END;
197
198 IF pResult = 'usernameExists'
199 THEN
200    RETURN;
201 END IF;
202
203 -- Additional code for cursor handling follows
204
205 END;
206
207 IF pResult = 'usernameExists'
208 THEN
209    RETURN;
210 END IF;
211
212 -- Additional code for cursor handling follows
213
214 END;
215
216 IF pResult = 'usernameExists'
217 THEN
218    RETURN;
219 END IF;
220
221 -- Additional code for cursor handling follows
222
223 END;
224
225 IF pResult = 'usernameExists'
226 THEN
227    RETURN;
228 END IF;
229
230 -- Additional code for cursor handling follows
231
232 END;
233
234 IF pResult = 'usernameExists'
235 THEN
236    RETURN;
237 END IF;
238
239 -- Additional code for cursor handling follows
240
241 END;
242
243 IF pResult = 'usernameExists'
244 THEN
245    RETURN;
246 END IF;
247
248 -- Additional code for cursor handling follows
249
250 END;
251
252 IF pResult = 'usernameExists'
253 THEN
254    RETURN;
255 END IF;
256
257 -- Additional code for cursor handling follows
258
259 END;
260
261 IF pResult = 'usernameExists'
262 THEN
263    RETURN;
264 END IF;
265
266 -- Additional code for cursor handling follows
267
268 END;
269
270 IF pResult = 'usernameExists'
271 THEN
272    RETURN;
273 END IF;
274
275 -- Additional code for cursor handling follows
276
277 END;
278
279 IF pResult = 'usernameExists'
280 THEN
281    RETURN;
282 END IF;
283
284 -- Additional code for cursor handling follows
285
286 END;
287
288 IF pResult = 'usernameExists'
289 THEN
290    RETURN;
291 END IF;
292
293 -- Additional code for cursor handling follows
294
295 END;
296
297 IF pResult = 'usernameExists'
298 THEN
299    RETURN;
300 END IF;
301
302 -- Additional code for cursor handling follows
303
304 END;
305
306 IF pResult = 'usernameExists'
307 THEN
308    RETURN;
309 END IF;
310
311 -- Additional code for cursor handling follows
312
313 END;
314
315 IF pResult = 'usernameExists'
316 THEN
317    RETURN;
318 END IF;
319
320 -- Additional code for cursor handling follows
321
322 END;
323
324 IF pResult = 'usernameExists'
325 THEN
326    RETURN;
327 END IF;
328
329 -- Additional code for cursor handling follows
330
331 END;
332
333 IF pResult = 'usernameExists'
334 THEN
335    RETURN;
336 END IF;
337
338 -- Additional code for cursor handling follows
339
340 END;
341
342 IF pResult = 'usernameExists'
343 THEN
344    RETURN;
345 END IF;
346
347 -- Additional code for cursor handling follows
348
349 END;
350
351 IF pResult = 'usernameExists'
352 THEN
353    RETURN;
354 END IF;
355
356 -- Additional code for cursor handling follows
357
358 END;
359
360 IF pResult = 'usernameExists'
361 THEN
362    RETURN;
363 END IF;
364
365 -- Additional code for cursor handling follows
366
367 END;
368
369 IF pResult = 'usernameExists'
370 THEN
371    RETURN;
372 END IF;
373
374 -- Additional code for cursor handling follows
375
376 END;
377
378 IF pResult = 'usernameExists'
379 THEN
380    RETURN;
381 END IF;
382
383 -- Additional code for cursor handling follows
384
385 END;
386
387 IF pResult = 'usernameExists'
388 THEN
389    RETURN;
390 END IF;
391
392 -- Additional code for cursor handling follows
393
394 END;
395
396 IF pResult = 'usernameExists'
397 THEN
398    RETURN;
399 END IF;
400
401 -- Additional code for cursor handling follows
402
403 END;
404
405 IF pResult = 'usernameExists'
406 THEN
407    RETURN;
408 END IF;
409
410 -- Additional code for cursor handling follows
411
412 END;
413
414 IF pResult = 'usernameExists'
415 THEN
416    RETURN;
417 END IF;
418
419 -- Additional code for cursor handling follows
420
421 END;
422
423 IF pResult = 'usernameExists'
424 THEN
425    RETURN;
426 END IF;
427
428 -- Additional code for cursor handling follows
429
430 END;
431
432 IF pResult = 'usernameExists'
433 THEN
434    RETURN;
435 END IF;
436
437 -- Additional code for cursor handling follows
438
439 END;
440
441 IF pResult = 'usernameExists'
442 THEN
443    RETURN;
444 END IF;
445
446 -- Additional code for cursor handling follows
447
448 END;
449
450 IF pResult = 'usernameExists'
451 THEN
452    RETURN;
453 END IF;
454
455 -- Additional code for cursor handling follows
456
457 END;
458
459 IF pResult = 'usernameExists'
460 THEN
461    RETURN;
462 END IF;
463
464 -- Additional code for cursor handling follows
465
466 END;
467
468 IF pResult = 'usernameExists'
469 THEN
470    RETURN;
471 END IF;
472
473 -- Additional code for cursor handling follows
474
475 END;
476
477 IF pResult = 'usernameExists'
478 THEN
479    RETURN;
480 END IF;
481
482 -- Additional code for cursor handling follows
483
484 END;
485
486 IF pResult = 'usernameExists'
487 THEN
488    RETURN;
489 END IF;
490
491 -- Additional code for cursor handling follows
492
493 END;
494
495 IF pResult = 'usernameExists'
496 THEN
497    RETURN;
498 END IF;
499
500 -- Additional code for cursor handling follows
501
502 END;
503
504 IF pResult = 'usernameExists'
505 THEN
506    RETURN;
507 END IF;
508
509 -- Additional code for cursor handling follows
510
511 END;
512
513 IF pResult = 'usernameExists'
514 THEN
515    RETURN;
516 END IF;
517
518 -- Additional code for cursor handling follows
519
520 END;
521
522 IF pResult = 'usernameExists'
523 THEN
524    RETURN;
525 END IF;
526
527 -- Additional code for cursor handling follows
528
529 END;
530
531 IF pResult = 'usernameExists'
532 THEN
533    RETURN;
534 END IF;
535
536 -- Additional code for cursor handling follows
537
538 END;
539
540 IF pResult = 'usernameExists'
541 THEN
542    RETURN;
543 END IF;
544
545 -- Additional code for cursor handling follows
546
547 END;
548
549 IF pResult = 'usernameExists'
550 THEN
551    RETURN;
552 END IF;
553
554 -- Additional code for cursor handling follows
555
556 END;
557
558 IF pResult = 'usernameExists'
559 THEN
560    RETURN;
561 END IF;
562
563 -- Additional code for cursor handling follows
564
565 END;
566
567 IF pResult = 'usernameExists'
568 THEN
569    RETURN;
570 END IF;
571
572 -- Additional code for cursor handling follows
573
574 END;
575
576 IF pResult = 'usernameExists'
577 THEN
578    RETURN;
579 END IF;
580
581 -- Additional code for cursor handling follows
582
583 END;
584
585 IF pResult = 'usernameExists'
586 THEN
587    RETURN;
588 END IF;
589
590 -- Additional code for cursor handling follows
591
592 END;
593
594 IF pResult = 'usernameExists'
595 THEN
596    RETURN;
597 END IF;
598
599 -- Additional code for cursor handling follows
599
600 END;
601
602 IF pResult = 'usernameExists'
603 THEN
604    RETURN;
605 END IF;
606
607 -- Additional code for cursor handling follows
608
609 END;
610
611 IF pResult = 'usernameExists'
612 THEN
613    RETURN;
614 END IF;
615
616 -- Additional code for cursor handling follows
617
618 END;
619
620 IF pResult = 'usernameExists'
621 THEN
622    RETURN;
623 END IF;
624
625 -- Additional code for cursor handling follows
626
627 END;
628
629 IF pResult = 'usernameExists'
630 THEN
631    RETURN;
632 END IF;
633
634 -- Additional code for cursor handling follows
635
636 END;
637
638 IF pResult = 'usernameExists'
639 THEN
640    RETURN;
641 END IF;
642
643 -- Additional code for cursor handling follows
644
645 END;
646
647 IF pResult = 'usernameExists'
648 THEN
649    RETURN;
650 END IF;
651
652 -- Additional code for cursor handling follows
653
654 END;
655
656 IF pResult = 'usernameExists'
657 THEN
658    RETURN;
659 END IF;
660
661 -- Additional code for cursor handling follows
662
663 END;
664
665 IF pResult = 'usernameExists'
666 THEN
667    RETURN;
668 END IF;
669
670 -- Additional code for cursor handling follows
671
672 END;
673
674 IF pResult = 'usernameExists'
675 THEN
676    RETURN;
677 END IF;
678
679 -- Additional code for cursor handling follows
680
681 END;
682
683 IF pResult = 'usernameExists'
684 THEN
685    RETURN;
686 END IF;
687
688 -- Additional code for cursor handling follows
689
690 END;
691
692 IF pResult = 'usernameExists'
693 THEN
694    RETURN;
695 END IF;
696
697 -- Additional code for cursor handling follows
698
699 END;
700
701 IF pResult = 'usernameExists'
702 THEN
703    RETURN;
704 END IF;
705
706 -- Additional code for cursor handling follows
707
708 END;
709
710 IF pResult = 'usernameExists'
711 THEN
712    RETURN;
713 END IF;
714
715 -- Additional code for cursor handling follows
716
717 END;
718
719 IF pResult = 'usernameExists'
720 THEN
721    RETURN;
722 END IF;
723
724 -- Additional code for cursor handling follows
725
726 END;
727
728 IF pResult = 'usernameExists'
729 THEN
730    RETURN;
731 END IF;
732
733 -- Additional code for cursor handling follows
734
735 END;
736
737 IF pResult = 'usernameExists'
738 THEN
739    RETURN;
740 END IF;
741
742 -- Additional code for cursor handling follows
743
744 END;
745
746 IF pResult = 'usernameExists'
747 THEN
748    RETURN;
749 END IF;
750
751 -- Additional code for cursor handling follows
752
753 END;
754
755 IF pResult = 'usernameExists'
756 THEN
757    RETURN;
758 END IF;
759
759
760 -- Additional code for cursor handling follows
761
762 END;
763
764 IF pResult = 'usernameExists'
765 THEN
766    RETURN;
767 END IF;
768
769 -- Additional code for cursor handling follows
770
771 END;
772
773 IF pResult = 'usernameExists'
774 THEN
775    RETURN;
776 END IF;
777
778 -- Additional code for cursor handling follows
779
780 END;
781
782 IF pResult = 'usernameExists'
783 THEN
784    RETURN;
785 END IF;
786
787 -- Additional code for cursor handling follows
788
789 END;
790
791 IF pResult = 'usernameExists'
792 THEN
793    RETURN;
794 END IF;
795
796 -- Additional code for cursor handling follows
797
798 END;
799
800 IF pResult = 'usernameExists'
801 THEN
802    RETURN;
803 END IF;
804
805 -- Additional code for cursor handling follows
806
807 END;
808
809 IF pResult = 'usernameExists'
810 THEN
811    RETURN;
812 END IF;
813
814 -- Additional code for cursor handling follows
815
816 END;
817
818 IF pResult = 'usernameExists'
819 THEN
820    RETURN;
821 END IF;
822
823 -- Additional code for cursor handling follows
824
825 END;
826
827 IF pResult = 'usernameExists'
828 THEN
829    RETURN;
830 END IF;
831
832 -- Additional code for cursor handling follows
833
834 END;
835
836 IF pResult = 'usernameExists'
837 THEN
838    RETURN;
839 END IF;
840
841 -- Additional code for cursor handling follows
842
843 END;
844
845 IF pResult = 'usernameExists'
846 THEN
847    RETURN;
848 END IF;
849
850 -- Additional code for cursor handling follows
851
852 END;
853
854 IF pResult = 'usernameExists'
855 THEN
856    RETURN;
857 END IF;
858
859 -- Additional code for cursor handling follows
860
861 END;
862
863 IF pResult = 'usernameExists'
864 THEN
865    RETURN;
866 END IF;
867
868 -- Additional code for cursor handling follows
869
870 END;
871
872 IF pResult = 'usernameExists'
873 THEN
874    RETURN;
875 END IF;
876
877 -- Additional code for cursor handling follows
878
879 END;
880
881 IF pResult = 'usernameExists'
882 THEN
883    RETURN;
884 END IF;
885
886 -- Additional code for cursor handling follows
887
888 END;
889
890 IF pResult = 'usernameExists'
891 THEN
892    RETURN;
893 END IF;
894
895 -- Additional code for cursor handling follows
896
897 END;
898
899 IF pResult = 'usernameExists'
900 THEN
901    RETURN;
902 END IF;
903
904 -- Additional code for cursor handling follows
905
906 END;
907
908 IF pResult = 'usernameExists'
909 THEN
910    RETURN;
911 END IF;
912
913 -- Additional code for cursor handling follows
914
915 END;
916
917 IF pResult = 'usernameExists'
918 THEN
919    RETURN;
920 END IF;
921
922 -- Additional code for cursor handling follows
923
924 END;
925
926 IF pResult = 'usernameExists'
927 THEN
928    RETURN;
929 END IF;
930
931 -- Additional code for cursor handling follows
932
933 END;
934
935 IF pResult = 'usernameExists'
936 THEN
937    RETURN;
938 END IF;
939
940 -- Additional code for cursor handling follows
941
942 END;
943
944 IF pResult = 'usernameExists'
945 THEN
946    RETURN;
947 END IF;
948
949 -- Additional code for cursor handling follows
950
951 END;
952
953 IF pResult = 'usernameExists'
954 THEN
955    RETURN;
956 END IF;
957
958 -- Additional code for cursor handling follows
959
960 END;
961
962 IF pResult = 'usernameExists'
963 THEN
964    RETURN;
965 END IF;
966
967 -- Additional code for cursor handling follows
968
969 END;
970
971 IF pResult = 'usernameExists'
972 THEN
973    RETURN;
974 END IF;
975
976 -- Additional code for cursor handling follows
977
978 END;
979
980 IF pResult = 'usernameExists'
981 THEN
982    RETURN;
983 END IF;
984
985 -- Additional code for cursor handling follows
986
987 END;
988
989 IF pResult = 'usernameExists'
990 THEN
991    RETURN;
992 END IF;
993
994 -- Additional code for cursor handling follows
995
996 END;
997
998 IF pResult = 'usernameExists'
999 THEN
1000    RETURN;
1001 END IF;
1002
1003 -- Additional code for cursor handling follows
1004
1005 END;
1006
1007 IF pResult = 'usernameExists'
1008 THEN
1009    RETURN;
1010 END IF;
1011
1012 -- Additional code for cursor handling follows
1013
1014 END;
1015
1016 IF pResult = 'usernameExists'
1017 THEN
1018    RETURN;
1019 END IF;
1020
1021 -- Additional code for cursor handling follows
1022
1023 END;
1024
1025 IF pResult = 'usernameExists'
1026 THEN
1027    RETURN;
1028 END IF;
1029
1030 -- Additional code for cursor handling follows
1031
1032 END;
1033
1034 IF pResult = 'usernameExists'
1035 THEN
1036    RETURN;
1037 END IF;
1038
1039 -- Additional code for cursor handling follows
1040
1041 END;
1042
1043 IF pResult = 'usernameExists'
1044 THEN
1045    RETURN;
1046 END IF;
1047
1048 -- Additional code for cursor handling follows
1049
1050 END;
1051
1052 IF pResult = 'usernameExists'
1053 THEN
1054    RETURN;
1055 END IF;
1056
1057 -- Additional code for cursor handling follows
1058
1059 END;
1060
1061 IF pResult = 'usernameExists'
1062 THEN
1063    RETURN;
1064 END IF;
1065
1066 -- Additional code for cursor handling follows
1067
1068 END;
1069
1070 IF pResult = 'usernameExists'
1071 THEN
1072    RETURN;
1073 END IF;
1074
1075 -- Additional code for cursor handling follows
1076
1077 END;
1078
1079 IF pResult = 'usernameExists'
1080 THEN
1081    RETURN;
1082 END IF;
1083
1084 -- Additional code for cursor handling follows
1085
1086 END;
1087
1088 IF pResult = 'usernameExists'
1089 THEN
1090    RETURN;
1091 END IF;
1092
1093 -- Additional code for cursor handling follows
1094
1095 END;
1096
1097 IF pResult = 'usernameExists'
1098 THEN
1099    RETURN;
1100 END IF;
1101
1102 -- Additional code for cursor handling follows
1103
1104 END;
1105
1106 IF pResult = 'usernameExists'
1107 THEN
1108    RETURN;
1109 END IF;
1110
1111 -- Additional code for cursor handling follows
1112
1113 END;
1114
1115 IF pResult = 'usernameExists'
1116 THEN
1117    RETURN;
1118 END IF;
1119
1120 -- Additional code for cursor handling follows
1121
1122 END;
1123
1124 IF pResult = 'usernameExists'
1125 THEN
1126    RETURN;
1127 END IF;
1128
1129 -- Additional code for cursor handling follows
1130
1131 END;
1132
1133 IF pResult = 'usernameExists'
1134 THEN
1135    RETURN;
1136 END IF;
1137
1138 -- Additional code for cursor handling follows
1139
1140 END;
1141
1142 IF pResult = 'usernameExists'
1143 THEN
1144    RETURN;
1145 END IF;
1146
1147 -- Additional code for cursor handling follows
1148
1149 END;
1150
1151 IF pResult = 'usernameExists'
1152 THEN
1153    RETURN;
1154 END IF;
1155
1156 -- Additional code for cursor handling follows
1157
1158 END;
1159
1160 IF pResult = 'usernameExists'
1161 THEN
1162    RETURN;
1163 END IF;
1164
1165 -- Additional code for cursor handling follows
1166
1167 END;
1168
1169 IF pResult = 'usernameExists'
1170 THEN
1171    RETURN;
1172 END IF;
1173
1174 -- Additional code for cursor handling follows
1175
1176 END;
1177
1178 IF pResult = 'usernameExists'
1179 THEN
1180    RETURN;
1181 END IF;
1182
1183 -- Additional code for cursor handling follows
1184
1185 END;
1186
1187 IF pResult = 'usernameExists'
1188 THEN
1189    RETURN;
1190 END IF;
1191
1192 -- Additional code for cursor handling follows
1193
1194 END;
1195
1196 IF pResult = 'usernameExists'
1197 THEN
1198    RETURN;
1199 END IF;
1200
1201 -- Additional code for cursor handling follows
1202
1203 END;
1204
1205 IF pResult = 'usernameExists'
1206 THEN
1207    RETURN;
1208 END IF;
1209
1210 -- Additional code for cursor handling follows
1211
1212 END;
1213
1214 IF pResult = 'usernameExists'
1215 THEN
1216    RETURN;
1217 END IF;
1218
1219 -- Additional code for cursor handling follows
1220
1221 END;
1222
1223 IF pResult = 'usernameExists'
1224 THEN
1225    RETURN;
1226 END IF;
1227
1228 -- Additional code for cursor handling follows
1229
1230 END;
1231
1232 IF pResult = 'usernameExists'
1233 THEN
1234    RETURN;
1235 END IF;
1236
1237 -- Additional code for cursor handling follows
1238
1239 END;
1240
1241 IF pResult = 'usernameExists'
1242 THEN
1243    RETURN;
1244 END IF;
1245
1246 -- Additional code for cursor handling follows
1247
1248 END;
1249
1250 IF pResult = 'usernameExists'
1251 THEN
1252    RETURN;
1253 END IF;
1254
1255 -- Additional code for cursor handling follows
1256
1257 END;
1258
1259 IF pResult = 'usernameExists'
1260 THEN
1261    RETURN;
1262 END IF;
1263
1264 -- Additional code for cursor handling follows
1265
1266 END;
1267
1268 IF pResult = 'usernameExists'
1269 THEN
1270    RETURN;
1271 END IF;
1272
1273 -- Additional code for cursor handling follows
1274
1275 END;
1276
1277 IF pResult = 'usernameExists'
1278 THEN
1279    RETURN;
1280 END IF;
1281
1282 -- Additional code for cursor handling follows
1283
1284 END;
1285
1286 IF pResult = 'usernameExists'
1287 THEN
1288    RETURN;
1289 END IF;
1290
1291 -- Additional code for cursor handling follows
1292
1293 END;
1294
1295 IF pResult = 'usernameExists'
1296 THEN
1297    RETURN;
1298 END IF;
1299
1300 -- Additional code for cursor handling follows
1301
1302 END;
1303
1304 IF pResult = 'usernameExists'
1305 THEN
1306    RETURN;
1307 END IF;
1308
1309 -- Additional code for cursor handling follows
1310
1311 END;
1312
1313 IF pResult = 'usernameExists'
1314 THEN
1315    RETURN;
1316 END IF;
1317
1318 -- Additional code for cursor handling follows
1319
1320 END;
1321
1322 IF pResult = 'usernameExists'
1323 THEN
1324    RETURN;
1325 END IF;
1326
1327 -- Additional code for cursor handling follows
1328
1329 END;
1330
1331 IF pResult = 'usernameExists'
1332 THEN
1333    RETURN;
1334 END IF;
1335
1336 -- Additional code for cursor handling follows
1337
1338 END;
1339
1340 IF pResult = 'usernameExists'
1341 THEN
1342    RETURN;
1343 END IF;
1344
1345 -- Additional code for cursor handling follows
1346
1347 END;
1348
1349 IF pResult = 'usernameExists'
1350 THEN
1351    RETURN;
1352 END IF;
1353
1354 -- Additional code for cursor handling follows
1355
1356 END;
1357
1358 IF pResult = 'usernameExists'
1359 THEN
1360    RETURN;
1361 END IF;
1362
1363 -- Additional code for cursor handling follows
1364
1365 END;
1366
1367 IF pResult = 'usernameExists'
1368 THEN
1369    RETURN;
1370 END IF;
1371
1372 -- Additional code for cursor handling follows
1373
1374 END;
1375
1376 IF pResult = 'usernameExists'
1377 THEN
1378    RETURN;
1379 END IF;
1380
1381 -- Additional code for cursor handling follows
1382
1383 END;
1384
1385 IF pResult = 'usernameExists'
1386 THEN
1387    RETURN;
1388 END IF;
1389
1390 -- Additional code for cursor handling follows
1391
1392 END;
1393
1394 IF pResult = 'usernameExists'
1395 THEN
1396    RETURN;
1397 END IF;
1398
1399 -- Additional code for cursor handling follows
1400
1401 END;
1402
1403 IF pResult = 'usernameExists'
1404 THEN
1405    RETURN;
1406 END IF;
1407
1408 -- Additional code for cursor handling follows
1409
1410 END;
1411
1412 IF pResult = 'usernameExists'
1413 THEN
1414    RETURN;
1415 END IF;
1416
1417 -- Additional code for cursor handling follows
1418
1419 END;
1420
1421 IF pResult = 'usernameExists'
1422 THEN
1423    RETURN;
1424 END IF;
1425
1426 -- Additional code for cursor handling follows
1427
1428 END;
1429
1430 IF pResult = 'usernameExists'
1431 THEN
1432    RETURN;
1433 END IF;
1434
1435 -- Additional code for cursor handling follows
1436
1437 END;
1438
1439 IF pResult = 'usernameExists'
1440 THEN
1441    RETURN;
1442 END IF;
1443
1444 -- Additional code for cursor handling follows
1445
1446 END;
1447
1448 IF pResult = 'usernameExists'
1449 THEN
1450    RETURN;
1451 END IF;
1452
1453 -- Additional code for cursor handling follows
1454
1455 END;
1456
1457 IF pResult = 'usernameExists'
1458 THEN
1459    RETURN;
1460 END IF;
1461
1462 -- Additional code for cursor handling follows
1463
1464 END;
1465
1466 IF pResult = 'usernameExists'
1467 THEN
1468    RETURN;
1469 END IF;
1470
1471 -- Additional code for cursor handling follows
1472
1473 END;
1474
1475 IF pResult = 'usernameExists'
1476 THEN
1477    RETURN;
1478 END IF;
1479
1480 -- Additional code for cursor handling follows
1481
1482 END;
1483
1484 IF pResult = 'usernameExists'
1485 THEN
1486    RETURN;
1487 END IF;
1488
1489 -- Additional code for cursor handling follows
1490
1491 END;
1492
1493 IF pResult = 'usernameExists'
1494 THEN
1495    RETURN;
1496 END IF;
1497
1498 -- Additional code for cursor handling follows
1499
1500 END;
1501
1502 IF pResult = 'usernameExists'
1503 THEN
1504    RETURN;
1505 END IF;
1506
1507 -- Additional code for cursor handling follows
1508
1509 END;
1510
1511 IF pResult = 'usernameExists'
1512 THEN
1513    RETURN;
1514 END IF;
1515
1516 -- Additional code for cursor handling follows
1517
1518 END;
1519
1520 IF pResult = 'usernameExists'
1521 THEN
1522    RETURN;
1523 END IF;
1524
1525 -- Additional code for cursor handling follows
1526
1527 END;
1528
1529 IF pResult = 'usernameExists'
1530 THEN
1531    RETURN;
1532 END IF;
1533
1534 -- Additional code for cursor handling follows
1535
1536 END;
1537
1538 IF pResult = 'usernameExists'
1539 THEN
1540    RETURN;
1541 END IF;
1542
1543 -- Additional code for cursor handling follows
1544
1545 END;
1546
1547 IF pResult = 'usernameExists'
1548 THEN
1549    RETURN;
1550 END IF;
1551
1552 -- Additional code for cursor handling follows
1553
1554 END;
1555
1556 IF pResult = 'usernameExists'
1557 THEN
1558    RETURN;
1559 END IF;
1560
1561 -- Additional code for cursor handling follows
1562
1563 END;
1564
1565 IF pResult = 'usernameExists'
1566 THEN
1567    RETURN;
1568 END IF;
1569
1570 -- Additional code for cursor handling follows
1571
1572 END;
1573
1574 IF pResult = 'usernameExists'
1575 THEN
1576    RETURN;
1577 END IF;
1578
1579 -- Additional code for cursor handling follows
1580
1581 END;
1582
1583 IF pResult = 'usernameExists'
1584 THEN
1585    RETURN;
1586 END IF;
1587
1588 -- Additional code for cursor handling follows
1589
1590 END;
1591
1592 IF pResult = 'usernameExists'
1593 THEN
1594    RETURN;
1595 END IF;
1596
1597 -- Additional code for cursor handling follows
1598
1599 END;
1600
1601 IF pResult = 'usernameExists'
1602 THEN
1603    RETURN;
1604 END IF;
1605
1606 -- Additional code for cursor handling follows
1607
1608 END;
1609
1610 IF pResult = 'usernameExists'
1611 THEN
1612    RETURN;
1613 END IF;
1614
1615 -- Additional code for cursor handling follows
1616
1617 END;
1618
1619 IF pResult = 'usernameExists'
1620 THEN
1621    RETURN;
1622 END IF;
1623
1624 -- Additional code for cursor handling follows
1625
1626 END;
1627
1628 IF pResult = 'usernameExists'
1629 THEN
1630    RETURN;
1631 END IF;
1632
1633 -- Additional code for cursor handling follows
1634
1635 END;
1636
1637 IF pResult = 'usernameExists'
1638 THEN
1639    RETURN;
1640 END IF;
1641
1642 -- Additional code for cursor handling follows
1643
1644 END;
1645
1646 IF pResult = 'usernameExists'
1647 THEN
1648    RETURN;
1649 END IF;
1650
1651 -- Additional code for cursor handling follows
1652
1653 END;
1654
1655 IF pResult = 'usernameExists'
1656 THEN
1657    RETURN;
1658 END IF;
1659
1660 -- Additional code for cursor handling follows
1661
1662 END;
1663
1664 IF pResult = 'usernameExists'
1665 THEN
1666    RETURN;
1667 END IF;
1668
1669 -- Additional code for cursor handling follows
1670
1671 END;
1672
1673 IF pResult = 'usernameExists'
1674 THEN
1675    RETURN;
1676 END IF;
1677
1678 -- Additional code for cursor handling follows
1679
1680 END;
1681
1682 IF pResult = 'usernameExists'
1683 THEN
1684    RETURN;
1685 END IF;
1686
1687 -- Additional code for cursor handling follows
1688
1689 END;
1690
1691 IF pResult = 'usernameExists'
1692 THEN
1693    RETURN;
1694 END IF;
1695
1696 -- Additional code for cursor handling follows
1697
1698 END;
1699
1700 IF pResult = 'usernameExists'
1701 THEN
1702    RETURN;
1703 END IF;
1704
1705 -- Additional code for cursor handling follows
1706
1707 END;
1708
1709
```

### 7.2.8 Database Teacher Details Entity

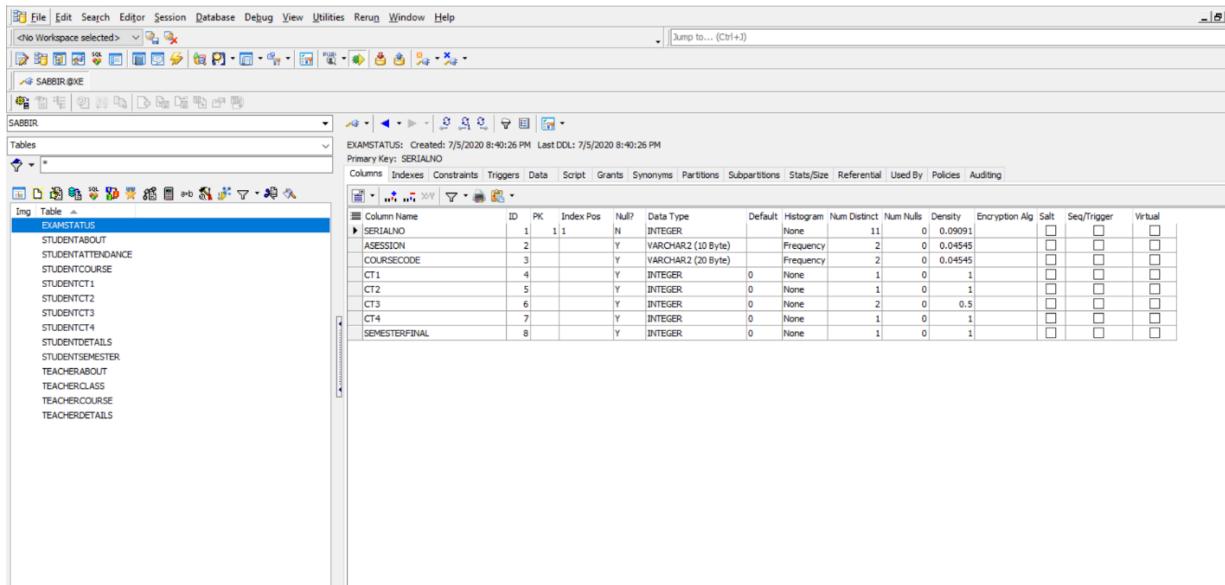


The screenshot shows the Oracle SQL Developer Schema Browser interface. The left sidebar lists various database objects like STUDENTABOUT, STUDENTATTENDANCE, STUDENTCOURSE, etc. The main panel displays the structure of the 'TEACHERDETAILS' table. The table has 13 columns:

Column Name	ID	PK	Index Pos	Null?	Data Type	Default	Histogram	Num Distinct	Num Nulls	Density	Encryption Alg	Salt	Sea/Trigger	Virtual
SERIALNO		1		Y	INTEGER	None		2	0	0.5				
DOJ		2		Y	VARCHAR2(20 Byte)	None		2	0	0.5				
NAME		3		Y	VARCHAR2(40 Byte)	None		2	0	0.5				
DOE		4		Y	VARCHAR2(20 Byte)	None		2	0	0.5				
BLOODGROUP		5		Y	VARCHAR2(20 Byte)	None		2	0	0.5				
IMAGE		6	1	Y	BLOB	None		0	0	0				
EMAIL		7		Y	VARCHAR2(65 Byte)	None		2	0	0.5				
PHONE		8		Y	VARCHAR2(15 Byte)	None		2	0	0.5				
FACULTY		9		Y	VARCHAR2(30 Byte)	None		1	0	1				
DEPT		10		Y	VARCHAR2(30 Byte)	None		2	0	0.5				
DESIGNATION		11		Y	VARCHAR2(30 Byte)	None		2	0	0.5				
USERNAME		12	1	N	VARCHAR2(30 Byte)	None		2	0	0.5				
PASSWORD		13		Y	VARCHAR2(30 Byte)	None		1	0	1				

Figure 7.2.8- Database Teacher Details Entity

### 7.2.9 Database Exam Details Entity



The screenshot shows the Oracle SQL Developer Schema Browser interface. The left sidebar lists various database objects like STUDENTABOUT, STUDENTATTENDANCE, STUDENTCOURSE, etc. The main panel displays the structure of the 'EXAMSTATUS' table. The table has 8 columns:

Column Name	ID	PK	Index Pos	Null?	Data Type	Default	Histogram	Num Distinct	Num Nulls	Density	Encryption Alg	Salt	Sea/Trigger	Virtual
SERIALNO	1	1	1	N	INTEGER	None		11	0	0.09091				
ASESSION	2			Y	VARCHAR2(10 Byte)	Frequency		2	0	0.04545				
COURSECODE	3			Y	VARCHAR2(20 Byte)	Frequency		2	0	0.04545				
CT1	4			Y	INTEGER	0	None	1	0	1				
CT2	5			Y	INTEGER	0	None	1	0	1				
CT3	6			Y	INTEGER	0	None	2	0	0.5				
CT4	7			Y	INTEGER	0	None	1	0	1				
SEMESTERFINAL	8			Y	INTEGER	0	None	1	0	1				

Figure 7.2.9- Database Exam Details Entity

#### **8.1 Purpose**

System implementation is the condition where the theoretical design is turned into a working system. The most crucial stage in achieving a new successful system is that it will work efficiently and effectively. System implementation is the critical phase of the undertaking when the hypothetical outline is transformed into the pragmatic framework. The principles organized in the execution are as per the following:

- Planning
- Training
- System testing and
- Changeover Planning

Arranging is the primary assignment in the framework usage. At the season of execution of any framework, individuals from various offices and framework investigation include. They are affirmed to the down to earth issue of controlling different exercises of individuals outside their own particular information preparing divisions.

The line supervisors controlled through an execution organizing council. The board thinks about thoughts, issues, and objections of client division, it should likewise consider:

- The ramifications of the system environment
- Self-determination and portion for usage assignments
- Consultation with associations and assets accessible
- Standby offices and channels of correspondence

Student Performance management system will implement student details user handle date details, separate login details time table details. It will be used to enter subject wise performance .This

application elaborate performance table generate term wise, consolidate report provide to the End user. To select starting date to end date generate reports at the time of activities.

#### **8.2 System Maintenance**

System maintenance conforms the system to its original requirements and enhancement adds to system capability by incorporating new requirements. The web based system maintenance is far more than finding mistakes. Provision must be made for environment changes, which may affect either the computer, or other parts of the computer based Systems. Such action is regularly called support. It incorporates both the change of the system capacities and the remedies of flaws, which emerge amid the activity of another system.

It might include the proceeding with the inclusion of an expansive extent of PC division recourses. The principle assignment might be to adjust existing system in an evolving situation.

Back up for the whole database documents are taken and put away gadgets like blaze drives, pen drives, and plates with the goal that it is conceivable to reestablish the system at the most punctual. In the event that there is a breakdown or crumple, at that point, the framework offers arrangement to reestablish database records. Putting away information in a different optional gadget prompts a successful and proficient keeps up with the system. The assigned individual has adequate information of the association's PC passed based framework to have the capacity to judge the importance of each proposed change.

Security of the system is maintained by providing login interface to the user. Only who has the login is and password can enter the system. Maintainability and reliability if the system is kept very thoroughly as all the records kept in the database have the backups and system can restore if there is power loss. Thus, maintenance changes the existing system, enhancement adds features to the existing system, and development replaces the existing system. It is an important part of system development that includes the activities which corrects errors in system design and implementation, updates the documents, and tests the data.

### **9.1 Conclusion**

In this work, the web based Student's Performance is developed using Java, Oracle for back-end and CSS, HTML, JavaScript for designing which is fully meet the system's goals.

This system conquers numerous impediments joined in participation, this system spares an extraordinary measure of time and lessens blunders which may happen amid participation estimation.

The system I have created is completely responsive which can be utilized as a part of mobile, tablets and distinctive working system. Some different advantages are,

- Automated and online for simple openness
- It is a dynamic and adaptable system
- This system overcome many limitations to show students performance.
- This system saves a great amount of time and reduces errors which may occur during performance calculation.
- It is exceptionally easy to use and convenient
- The records can be accessible in the provoke and a prompt.

### **9.2 Scope for Future Development**

I will make some future improvement in my student attendance management system project by using new technology and add attractive feature in order to make more advanced and increase its reliability and effectiveness. I will make some future change in my undertaking by utilizing new apparatuses and advances with a specific end goal to make further developed and increment its unwavering quality and adequacy.

The following are the future scope for the project:

- To create admin module for maintaining overall system.
- Individual Attendance system with photo using Student login.
- Add new feature to represent the students co-curricular activities performance.

---

## BIBLIOGRAPHY

---

- <https://www.spring.io>
- <https://www.javatpoint.com>
- <https://www.baeldung.com/spring-boot>
- <https://www.oracletutorial.com>
- <https://www.w3resource.com>
- <https://getbootstrap.com>
- <https://www.w3schools.com/>
- <https://www.tutorialspoint.com/>
- <https://www.javatpoint.com/>