

A
Major Project Report
ON
“Web-Based Management System with AWS”

Submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology

In

Computer Science Engineering



By

Rabecca Fatima (PU-005161911A12)
Aasim Ahsan (PU-003161911A12)

UNDER THE GUIDANCE OF
Mr. Anjul Rai
Asst. Professor
SORT, IBM ICE

People's University, Bhopal, (M.P.)

People's University, Bhopal

School of Research & Technology

Department of Computer Science Engineering



DECLARATION

We hereby declare that the work, which is being presented as the Major Project Report entitled "**Web-Based Management System with AWS**" in partial fulfilment of the requirements for the award of degree of **Bachelor of Technology in Computer Science Engineering**, submitted in the department of Computer Science, IBM-ICE is our original work and the project has not formed the basis for the award of any other degree, diploma, fellowship or any other similar titles.

Place: **Rabeca Fatima (PU-005161911A12)**

Date: **Aasim Ahsan (PU-003161911A12)**

People's University, Bhopal

School of Research & Technology

Department of Computer Science Engineering



CERTIFICATE

This is to certify that Major Project entitled "**“Web-Based Management System with AWS”**" submitted by **Rabecca Fatima (PU-00516191A12) & Aasim Ahsan (PU-003161911A12)**, has been carried out under my guidance / supervision. The Major Project report is approved for submission towards partial fulfilment for the award of the degree of **Bachelor of Technology in Computer Science Engineering**.

It is further certified that this work has not been submitted for the award of any other degree of diploma.

Guided By:

Mr. Anjul Rai
Assistant Professor
IBM-ICE, SORT

Approved By:

Mr. Ankit
Dwivedi
Head of
Department
IBM-ICE, SORT

Forwarded By:

Dr. Adarsh Sachdeva
Principal
SORT

People's University, Bhopal
School of Research & Technology
Department of Computer Science Engineering



CERTIFICATE OF APPROVAL

This is to certify that the project titled "**Web-Based Management System with AWS**" is the bona fide work carried out by **Rabecca Fatima (PU-005161911A12) & Aasim Ahsan (PU-003161911A12)**, the students of B Tech (CSE) with specialization in Cloud Computing & Virtualization of SORT, People's University Bhopal during the academic year 2022-23, in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology (Computer Science and Engineering)

Internal Examiner

Date:.....

External Examiner

Date:.....

ABSTRACT

It is challenging for educational institutions or colleges to maintain the records of students using a manual method because the information is dispersed, and often redundant, and gathering the pertinent information can take a lot of time. Using this project, all of these issues can be resolved. The Web Based Student Management System (WBSMS) built using AWS Beanstalk would be a web-based application that allows users to manage and organize student information. This could include storing and updating student profiles, tracking student progress and performance, and generating reports on student activity. Beanstalk would provide the infrastructure necessary to host and run the application, while also handling deployment, scaling, and monitoring. It would eliminate the need for a manual system, which can be scattered and redundant and can be time-consuming to collect relevant information. Additionally, other AWS services such as RDS, S3 and DynamoDB could be integrated to handle data storage and management tasks. Overall, this project would provide an easy-to-use and efficient solution for managing student information in a school or educational institution.

Acknowledgement

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success. We are grateful to our project guide **Asst. Prof. Anjul Rai** for the guidance, inspiration and constructive suggestions that help us in the preparation of this project. And we also thank our mentor **Asst. Prof. Aseem Bhandari** who took a step forward in correcting our mistakes and learning to not repeat them later. We also thank our colleagues who have helped in successful completion of the project, without their constant cooperation and brilliant efforts this project would have been a failure, irrespective of the obstructions they faced, they didn't give up.

We would also like to thank **Dr. Adarsh Sachdeva**, principal for whole hearted support. We also wish to express our gratitude to **Mr. Ankit Dwivedi, HOD (IBM-ICE)** for his kind hearted support.

We are also grateful to our teachers for their constant support and guidance. We also wish to express our indebtedness to our parents as well as our family members for their blessings and support always helped me to face the challenges ahead

At the end we would like to express our sincere thanks to all our friends and others who helped us directly or indirectly during this project work.

Rabecca Fatima (PU-005161911A12)

Aasim Ahsan (PU-003161911A12)

TABLE OF CONTENTS

Contents

<i>DECLARATION</i>	I
<i>CERTIFICATE</i>	II
<i>CERTIFICATE OF APPROVAL</i>	III
ABSRACT	IV
ACKNOWLEDGEMENT	V
TABLE OF CONTENTS	2-1
.....	2-3
LIST OF FIGURE	2-3
CHAPTER 1 INTRODUCTION	2-4
1.1 STUDENT MANAGEMENT	2-4
1.2 AWS ELASTIC BEANSTALK	2-6
1.3 PROBLEM DEFINITION.....	2-7
1.4 PROPOSED SOLUTION	2-7
CHAPTER 2 LITERATURE REVIEW	2-8
2.1 EXISTING SYSTEM.....	2-9
2.2 STUDENT MANAGEMENT	2-10
2.2.1 <i>Student Management Techniques</i>	2-10
CHAPTER 3 REQUIREMENT & ANALYSIS	3-11
3.1 REQUIREMENTS FOR DEPLOYMENT.....	3-11
3.1.1 <i>Software Requirement</i>	3-11
3.1.2 <i>Hardware Requirement (Processor RAM Disk Space)</i>	3-13
3.2 REQUIREMENTS FOR ACCESSING	3-13
3.2.1 <i>Software Requirement</i>	3-13
3.2.2 <i>Hardware Requirement (Processor RAM Disk Space)</i>	3-13
CHAPTER 4 DESIGN	4-14
4.1 TECHNOLOGY SELECTION.....	4-14
4.2 USER INTERFACE DESIGN.....	4-17
CHAPTER 5 USE CASE REPORTS	4-20
5.1 ADMINISTRATOR:	4-20
BASIC FLOW: ①	4-21
ALTERNATIVE FLOW: ②	4-21

<i>Use Case Report- Login into the website</i>	4-21
PRECONDITIONS: ②	4-21
BASIC FLOW: ②	4-22
PRECONDITIONS: ②	4-23
BASIC FLOW: ②	4-23
5.2 STUDENT USE CASE :	4-24
BASIC FLOW: ②	4-24
USE CASE REPORT- REGISTER STUDENT ON WEBSITE.....	4-25
BASIC FLOW: ②	4-25
USE CASE REPORT- LOGIN INTO THE SYSTEM.....	4-26
PRECONDITIONS: ②	4-26
BASIC FLOW: ②	4-26
5.3 DEPLOYMENT ON ELASTIC BEANSTALK	4-27
5.4 SNAPSHOT.....	4-41
INDEX PAGE	4-41
ADMIN LOGIN.....	4-42
<i>After Login -> Add New Student.....</i>	4-43
<i>when save clicked</i>	4-44
<i>Add New Course.....</i>	4-44
<i>Add New Branch</i>	4-45
<i>branch saved.....</i>	4-45
<i>Add New Subjects according to course , branch ,semester ,year.....</i>	4-46
<i>Display all registered students.....</i>	4-47
<i>Display result of all students:.....</i>	4-48
<i>Student Register.....</i>	4-49
<i>Successfully registered.....</i>	4-50
<i>Student Login</i>	4-51
<i>Student Home page</i>	4-52
<i>View Result</i>	4-53
<i>Update student details.....</i>	4-54
<i>Update Student result if any(only done by the admin)</i>	4-55
5.5 FLOWCHART	4-56
SCOPE OF THE PROJECT	4-57
BIBLIOGRAPHY	4-58

List of Figure

Figure 5.1 Databases	47
Figure 5.2 DB Login.....	48
Figure 5.3 Project Flowchart	64

Chapter 1 INTRODUCTION

1.1 Student Management

Student management refers to the process of organizing, managing, and keeping track of the information and activities related to students in an educational institution. It involves a wide range of tasks such as maintaining student records, tracking student progress, managing student schedules and assignments, and generating reports on student activity. Student management systems can be either manual or computerized. Manual systems typically involve paper records and spreadsheets, while computerized systems use specialized software to manage and organize student data. A computerized system can provide many benefits over manual systems, such as improved data accuracy, better reporting capabilities, and more efficient data sharing between different departments and staff members. The design and implementation of a comprehensive student information system and user interface is to replace the current paper records [1]. College Staff are able to directly access all aspects of a student's academic progress through a secure, online interface embedded in the college's website. The system utilizes user authentication, displaying only information necessary for an individual's duties. Additionally, each sub-system has authentication allowing authorized users to create or update information in that subsystem. All data is thoroughly reviewed and validated on the server before actual record alteration occurs. In addition to a staff user interface, the system plans for student user interface, allowing users to access information and submit requests online thus reducing processing time. All data is stored securely on SQL servers managed by the college administrator and ensures highest possible level of security. The system features a complex logging system to track all users access and ensure conformity to data access guidelines and is expected to increase the efficiency of the college's record management thereby decreasing the work hours needed to access and deliver student records to users.

Previously, the college relied heavily on paper records for this initiative. While paper records are a traditional way of managing student data there are several drawbacks to this method. First, to convey information to the students it should be displayed on the notice board and the student has to visit the notice board to check that information. It takes a very long time to convey the information to the student. Paper records are difficult to manage and track. The physical exertion required to retrieve, alter, and re-file the paper records are all non-value added activities

Applications

A Student Management System (SMS) application is a software tool designed to help educational institutions manage and organize student information. It typically includes a range of features such as:

- Student Information Management: Allows administrators to store and update student profiles, including personal information, contact details, and academic records.
- Attendance Management: Allows teachers to track student attendance and generate reports on attendance patterns.
- Grading and Assessment: Allows teachers to track student progress, record grades, and generate reports on student performance.
- Scheduling: Allows administrators to create class schedules and assign teachers to classes.
- Communication: Allows teachers and administrators to communicate with students and parents through email, messaging, or other tools.
- Reporting: Generates reports on student activity and performance, which can be used for analysis and planning.
- Mobile accessibility: SMS application can be accessed from mobile devices which makes it easy for students, teachers, and admin to keep track of their data and activities.
- Integration with other systems: SMS can be integrated with other systems like payment gateway, library management system, and more.

A SMS application can be accessed by administrators, teachers, and students, and can be used to streamline and automate many of the tasks involved in managing student information, making it more efficient and accurate.

1.2 AWS Elastic Beanstalk

AWS Elastic Beanstalk is a fully managed service provided by Amazon Web Services (AWS) that makes it easy to deploy, run, and scale web applications and services. It abstracts away many of the underlying infrastructure details, making it simple for developers to quickly deploy and run their applications without worrying about provisioning servers or configuring load balancers.

With Elastic Beanstalk, developers can simply upload their application code and Beanstalk will automatically handle the deployment, scaling, and monitoring of the application. Beanstalk supports a wide variety of programming languages and frameworks, including Java, .NET, PHP, Node.js, Python, Ruby, and Go. It also integrates with other AWS services such as Amazon RDS for database management, Amazon S3 for storage, and Amazon CloudWatch for monitoring.

Additionally, Beanstalk allows you to easily configure and manage the underlying infrastructure of your application, such as the number of EC2 instances, the size of the instances, and the load balancer settings. It also provides monitoring and logging features, allowing you to track the performance of your application and troubleshoot any issues that arise.

Overall, Elastic Beanstalk is a powerful tool for deploying and managing web applications and services, making it easy for developers to focus on building great applications without worrying about the underlying infrastructure.

1.3 Problem Definition

The problem that a Student Management faces is the difficulties that educational institutions face in managing and organizing student information. The following are some specific problems that an SMS can address:

1. Dispersed and redundant information: Without a centralized system, student information is often stored in multiple locations, making it difficult to access and maintain. This can lead to inconsistencies and errors in student records.
2. Time-consuming data collection: Gathering relevant information about a student, such as attendance records, grades, and contact information, can take a lot of time when using manual methods.
3. Inefficient communication: Without an SMS, it can be difficult for teachers, administrators, and parents to communicate effectively.
4. Limited reporting capabilities: Without an SMS, it can be difficult to generate reports on student activity and performance, which are necessary for analysis and planning.
5. Difficulty in handling a large number of students: With an increasing number of students, it is becoming increasingly difficult for educational institutions to manage student information manually.

1.4 Proposed Solution

Using AWS Elastic Beanstalk to develop a Student Management System (SMS) can provide a solution to the difficulties that educational institutions face in managing and organizing student information. Some of the specific ways that Beanstalk can solve these problems include:

1. Centralized and Automated Management: Beanstalk provides a platform for deploying and managing web applications, which can be used to create a centralized system for managing student information. This eliminates the need for manual data entry and makes it easy to access and update student records.
2. Scalability: Beanstalk automatically handles scaling and load balancing, ensuring that the SMS can handle a large number of users, even during peak usage. This is especially important for larger educational institutions.

3. Integration with other AWS services: Beanstalk can be integrated with other AWS services such as RDS for database management, S3 for storage, and CloudWatch for monitoring. This allows for robust data storage and management capabilities.
4. Easy Deployment and Management: Beanstalk makes it easy to deploy, run and manage web applications, which allows developers to focus on building the SMS application rather than managing the underlying infrastructure.
5. Monitoring and Logging: Beanstalk provides monitoring and logging features, allowing administrators to track the performance of the SMS and troubleshoot any issues that arise.
6. Mobile accessibility: Beanstalk allows for mobile accessibility of the SMS, making it easy for students, teachers, and admin to access student information and perform tasks from anywhere.

Overall, AWS Elastic Beanstalk provides a solution to the difficulties in managing student information by providing a platform for building a centralized, scalable and easy to manage SMS.

Chapter 2 LITERATURE REVIEW

Pressman, Roger S.,Software Engineering “A Practitioner’ s Approach”, FifthEdition, McGraw-Hill, 2000. Software Engineering by K.K Aggarwal and Yogesh Singh

In India there are many academic institutions. But very few institutions are modernized and use software to manage their day to day work. In a city like Bengaluru there are around 1000 schools, more than 300 pre-university colleges and degree colleges. Most of these academic institutions still relay on traditional way of management which mainly involves paper-work, much of human effort.

The students, who are admitted to those institutions which are dependent on traditional way of managing things, have to struggle a lot just to get a certificate or any other documents. Also the administrations face difficulty in maintaining all the records, tracking the records and fetching the record of their interest in time. The administrations of those institutions also have to employ a number of employees just to maintain the records required to manage and support their daily work. Some of the universities like PESIT and Christ University in Bengaluru have their own web application to address the previously mentioned issues.

The web application that is being used by these and many other institutions have the following features and functionalities such as, Login/Sign Up, Dashboard, Viewing of results, attendance, courses, time table, assignments and student's progress, upload/download documents and notifications.

2.1 Existing System

Existing systems for student management typically fall into one of two categories: manual systems or computerized systems.

Manual systems typically involve paper records and spreadsheets, where student information is recorded by hand and stored in physical filing cabinets or folders. These systems can be time-consuming, prone to errors, and difficult to access and update.

Computerized systems, on the other hand, use specialized software to manage and organize student data. These systems can include features such as storing and updating student profiles, tracking student progress and performance, and generating reports on student activity. They can also be integrated with other systems such as payment gateways, library management systems, and more. Computerized systems can provide many benefits over manual systems, such as improved data accuracy, better reporting capabilities, and more efficient data sharing between different departments and staff members.

Some existing systems that are available in the market are:

1. Student Information System (SIS)
2. Learning Management System (LMS)
3. Campus Management System (CMS)
4. Student Relationship Management (SRM)

5. Student Data Management System (SDMS)

These systems can be installed on-premise or can be hosted on cloud platforms like AWS, Azure, and Google Cloud Platform.

It's important to evaluate and compare the different existing systems based on the specific needs and requirements of the educational institution to find the most suitable one for their needs.

2.2 Student Management

An SMS can provide many benefits over manual systems, such as improved data accuracy, better reporting capabilities, and more efficient data sharing between different departments and staff members. It can also provide mobile accessibility, which makes it easy for students, teachers, and administrators to access student information and perform tasks from anywhere.

AWS Elastic Beanstalk is a fully managed service provided by Amazon Web Services (AWS) that makes it easy to deploy, run, and scale web applications and services. It abstracts away many of the underlying infrastructure details, making it simple for developers to quickly deploy and run their SMS applications without worrying about provisioning servers or configuring load balancers. Beanstalk supports a wide variety of programming languages and frameworks, and it also integrates with other AWS services such as Amazon RDS for database management, Amazon S3 for storage, and Amazon CloudWatch for monitoring.

2.2.1 Student Management Techniques

There are several techniques that can be used to effectively manage a Student Management System (SMS):

- **Centralized Database:** A centralized database is a key component of an SMS, it allows for the storage, management and easy retrieval of student information.
- **Automated Data Entry:** Automated data entry can significantly reduce the time and effort required to enter and update student information. This can be achieved through the use of barcode scanners, RFID tags, or other automated data entry methods.

- Reporting and Analytics: An SMS should include advanced reporting and analytics capabilities that allow administrators to track student progress, generate reports on student activity and performance, and make data-driven decisions.
- Communication Tools: An SMS should include tools that allow teachers, administrators, and parents to communicate effectively, such as email, messaging, and social media.
- Mobile Accessibility: An SMS should be accessible from mobile devices, making it easy for students, teachers, and administrators to access student information and perform tasks from anywhere.
- Data Security: SMS should have proper data security mechanisms in place to protect student information from unauthorized access, alteration or deletion.
- Integration: An SMS should be able to integrate with other systems such as payment gateways, library management systems, and more.

Chapter 3 REQUIREMENT & ANALYSIS

3.1 Requirements for Deployment

For the deployment of the project following requirements need to be satisfied.

3.1.1 Software Requirement

- APACHE Tomcat Server 9.0

Apache Tomcat is an open-source Java Servlet Container developed by the Apache Software Foundation. Tomcat version 9.0 is the latest version and it requires JDK 8 or later. It supports the latest Java Servlet, JavaServer Pages, and Java Expression Language standards, as well as WebSocket, and JASPIC. It also includes many improvements in security, performance, and stability.

- **MYSQL Workbench 8.0 CE**

MySQL Workbench is a visual database design tool developed by MySQL. It is available in both Community Edition (CE) and Enterprise Edition (EE) versions. MySQL Workbench 8.0 CE is the latest version of the Community Edition, and it is compatible with MySQL Server versions 8.0 and later. It provides a visual and user-friendly interface for designing, modeling, and manipulating databases, and includes features such as a SQL editor, schema and data modeling tools, and a variety of administration and management tools. Additionally, it allows to perform database migrations, reverse engineering and forward engineering.

- **Eclipse IDE for Java EE Developers**

Eclipse IDE for Java EE Developers is a version of the Eclipse Integrated Development Environment (IDE) that is specifically tailored for developing Java Enterprise Edition (Java EE) applications. It is built on top of the standard Eclipse platform and includes additional tools and plugins for developing Java EE projects, such as support for JavaServer Faces (JSF), JavaServer Pages (JSP), Enterprise JavaBeans (EJB), and Java Persistence API (JPA). It also includes a built-in web server (such as Apache Tomcat or Glassfish), as well as support for debugging, source code management, and other development tasks. Additionally, it allows to integrate with other tools and frameworks like maven, gradle and spring boot.

- **AWS Elastic Beanstalk**

AWS Elastic Beanstalk is an orchestration service offered by Amazon Web Services for deploying applications which orchestrates various AWS services, including EC2, S3, Simple Notification Service (SNS), CloudWatch, autoscaling, and Elastic Load Balancers.[2] Elastic Beanstalk provides an additional layer of abstraction over the bare server and OS; users instead see a pre-built combination of OS and platform, such as "64bit Amazon Linux 2014.03 v1.1.0 running Ruby 2.0 (Puma)" or "64bit Debian jessie

v2.0.7 running Python 3.4 (Preconfigured - Docker)".[3] Deployment requires a number of components to be defined: an 'application' as a logical container for the project, a 'version' which is a deployable build of the application executable, a 'configuration template' that contains configuration information for both the Beanstalk environment and for the product.[4] Finally an 'environment' combines a 'version' with a 'configuration' and deploys them.[3] Executables themselves are uploaded as archive files to S3 beforehand and the 'version' is just a pointer to this

3.1.2 Hardware Requirement (Processor RAM Disk Space)

- Laptop or PC with Intel i5 processor or equivalent or higher (8 GB RAM) or higher

3.2 Requirements for Accessing

To access the project any user need to have the following software and hardware requirements.

3.2.1 Software Requirement

- APACHE Tomcat Server
- MYSQL Workbench 8.0 CE
- Eclipse IDE for Java EE Developers
- AWS Elastic Beanstalk

3.2.2 Hardware Requirement (Processor RAM Disk Space)

- Laptop or PC with Intel i5 processor or equivalent or higher (8 GB RAM) or higher

Chapter 4 DESIGN

4.1 Technology Selection

Java is divided into two parts i.e. Core Java (J2SE) and Advanced Java (JEE). The core Java part covers the fundamentals (data types, functions, operators, loops, thread, exception handling, etc.) of the Java programming language. It is used to develop general purpose applications. Whereas Advanced Java covers the standard concepts such as database connectivity, networking, Servlet, web-services, etc. It is a part of Java programming language. It is an advanced technology or advance version of Java specially designed to develop web-based, network-centric or enterprise applications. It includes the concepts like Servlet, JSP, JDBC, RMI, Socket programming, etc. It is a specialization in specific domain.

Most of the applications developed using advance Java uses tow-tier architecture i.e. Client and Server. All the applications that runs on Server can be considered as advance Java applications

Apache Tomcat (called "Tomcat" for short) is a free and open-source implementation of the Jakarta Servlet, Jakarta Expression Language, and WebSocket technologies.[2] It provides a "pure Java" HTTP web server environment in which Java code can also run. Thus it is a Java web application server, although not a full JEE application server.

Tomcat is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation, released under the Apache License 2.0 license.

Tomcat started off as a servlet reference implementation by James Duncan Davidson, a software architect at Sun Microsystems. He later helped make the project open-source and played a key role in its donation by Sun Microsystems to the Apache Software Foundation.[10] The Apache Ant software build automation tool was developed as a side-effect of the creation of Tomcat as an open source project.

Davidson had initially hoped that the project would become open-sourced and, since many open-source projects had O'Reilly books associated with them featuring an animal on the cover, he wanted to name the project after an animal. He came up with Tomcat since he reasoned the animal represented something that could fend for itself. Although the tomcat was already in use for another O'Reilly title,[11] his wish to see an animal cover eventually came true when O'Reilly published their Tomcat book with a snow leopard on the cover in 2003.[12]

MySQL Workbench is a unified visual tool for database architects, developers, and DBAs. MySQL Workbench provides data modeling, SQL development, and comprehensive administration tools for server configuration, user administration, backup, and much more. MySQL Workbench is available on Windows, Linux and Mac OS X.

MySQL Enterprise Edition includes the most comprehensive set of advanced features, management tools and technical support to achieve the highest levels of MySQL scalability,

security, reliability, and uptime. It reduces the risk, cost, and complexity in developing, deploying, and managing business-critical MySQL applications.

AWS Elastic Beanstalk is an orchestration service offered by Amazon Web Services for deploying applications which orchestrates various AWS services, including EC2, S3, Simple Notification Service (SNS), CloudWatch, autoscaling, and Elastic Load Balancers.[2] Elastic Beanstalk provides an additional layer of abstraction over the bare server and OS; users instead see a pre-built combination of OS and platform, such as "64bit Amazon Linux 2014.03 v1.1.0 running Ruby 2.0 (Puma)" or "64bit Debian jessie v2.0.7 running Python 3.4 (Preconfigured - Docker)".[3] Deployment requires a number of components to be defined: an 'application' as a logical container for the project, a 'version' which is a deployable build of the application executable, a 'configuration template' that contains configuration information for both the Beanstalk environment and for the product.[4] Finally an 'environment' combines a 'version' with a 'configuration' and deploys them.[3] Executables themselves are uploaded as archive files to S3 beforehand and the 'version' is just a pointer to this.[3]

JSON (JavaScript Object Notation, pronounced /'dʒeɪsən/; also /'dʒei sɒn/) is an open standard file format and data interchange format that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and arrays (or other serializable values). It is a common data format with diverse uses in electronic data interchange, including that of web applications with servers.

JSON is a language-independent data format. It was derived from JavaScript, but many modern programming languages include code to generate and parse JSON-format data. JSON filenames use the extension .json. Any valid JSON file is a valid JavaScript (.js) file, even though it makes no changes to a web page on its own.

Douglas Crockford originally specified the JSON format in the early 2000s.[1] He and Chip Morningstar sent the first JSON message in April 2001.

4.2 User Interface Design

The UI designed so far is coded using Eclipse IDE which is then deployed by the AWS Elastic Beanstalk. It contains the following:

- Online registration of students
- Maintenance of student records
- Searching student records
- Displaying student records

Users Views:

- Administrator
- Student

Login module: Login module will help in authentication of user accounts .Users who have valid login id and password can only login into their respective accounts.

Search module: Suppose there are hundreds of students and from this we have to search a particular student and we know the name of the student .In manual system it is a tedious task though we know the name of the student, but using this module we can easily search the student by specifying the name of the student in the search criteria. Thus this module will help the administrator in searching the student with various criteria easily.

Registration Module and Account Management: This module will help the student get registered from anywhere if internet is present .This module will really simplify the task of on paper registration. Also after successful registration the user can update information.

User Management: This module will help the administrator in enabling/disabling a user account and updating user information as required.

Purpose of project is to maintain details of the students such as storing information about:

- Student enrollment no.
- Student password
- Student name
- Student DOB
- Student mailing address
- Student father's name
- Student Gender
- Student contact no.
- Student email
- City
- Country

Definitions, Acronyms and Abbreviations :

- **Personal details:** Details of student such as user id, phone number, address, image, resume, e-mail address etc.
- **Contact details:** Details of contact associated with the student.

- **SRS:** System requirement Specification
- **WWW:** World Wide Web
- **Administrator:** A Login Id representing the user is an administrator & can access all the records details

Product Functions :

There are two different users who will be using this product:

- Administrator who can view and edit the details of any students.
 - Students who can view their details as well as they can edit their details.
1. The features that are available to the Administrator are:
 - An Administrator can login into the system and perform any of the available operations.
 - Can add student.
 - Can edit student information to the database.
 - Can make search for a specific student.
 - Can access all the details of the student.
 2. The features that are available to the student are:
 - Student can login into the system and can perform any of the available options.
 - Can view his/her personal details.
 - Can edit his/her personal details

- Can view & download his/her result.

Assumptions & dependencies

- Administrator is created in the system already.
- Roles and tasks are predefined.
- Student roll no is provided by the admin.

CHAPTER 5 Use Case Reports

5.1 Administrator:

Responsible for managing student details.

Use-case: Login into the website

Goal in context: Gain access to the website

Brief Description:

This use case is used when the administrator wants to access the website to enable/add/update the personal details of the student.

Preconditions: The Administrator must be logged onto the website in order for this use case to begin.

Basic Flow: □

- The Website prompts the administrator for the user name and password. □
- The Administrator enters the user name and password.
- The Website verifies the password and sets the user's authorization. □
- The Administrator is given access to the Website to perform his tasks.

Alternative Flow: □

The administrator enters invalid username and password then he will not be allowed to enter the website.

Post conditions:

The website state is unchanged by this use case.

Use Case Report- Login into the website

Use Case : Display student details

Goal in context: View the details of a student

Brief Description:

This use case is used when the administrator wants to view the personal details of the students already existing in the database on the screen.

Preconditions: □

- The Administrator must be logged into the system in order for this use case to begin □
- The details of the student must pre-exist in the database □
- The student id must be entered correctly.

Basic Flow: □

The Administrator logs onto the System. □

The Administrator search the student from following keys:-

- Student enrollment no.
- course
- branch

The System prompts for the student detail from one of the above keys.

□ The student details are displayed on the screen.

Alternative Flow:

Student Not Found If in the Display a student sub-flows, a student with the specified id number does not exist, The system displays an error message.

The Administrator can then enter a different id number or cancel the operation, at which point the use case ends.

Post conditions:

- The student details are displayed on the screen already existing in the system.
- The state of the system remains unchanged.

Use Case : Edit student details

Goal in context: Edit the details of a student Brief Description:

This use case is used when the administrator wants to edit the personal details of the himself/herself already existing in the database.

Preconditions: □

- The Administrator must be logged into the system in order for this use case to begin. □
- The details of the student must pre-exist in the database

Basic Flow: □

The Administrator logs onto the System. □

The Administrator can edit following keys:-

- Name
- Fathers Name
- Gender
- DOB
- Contact no
- Country
- City
- Email
- Address

The Website updates the database according to edited details.

- The student details are edited in the database.

Alternative Flow: There is no alternative flow of this use case diagram.

Post conditions:

The student details get updated in the database.

5.2 Student Use Case :

Goal in context: Registration of a student

Brief Description: This use case is used when the student register himself/herself in the database online.

Preconditions: The Student must accessed the website in order for this use case to begin. The user id must be unique and entered correctly.

Basic Flow:

The Student enters into the website.

The student view & modify his/her details from the following keys:-

- Student id
- password
- name
- Gender
- DOB
- Contact no

- Country
- City
- Email
- Contact
- Address

The System details are added to the database.

The student details are displayed on the screen.

Alternative Flow:

User ID not unique: if the user id entered is not unique then it will show an error message.

Post conditions: The student get registered on the website and to login into that particular the administrator must enable it.

Use Case Report- Register student on website

Use-case: Login into the website

Goal in context: Gain access to the website

Brief Description: This use case is used when the student wants to access the website.

Preconditions:

The Administrator must enabled the particular student onto the website in order for this use case to begin.

Basic Flow: □

- The website prompts the student for the user name and password. □

- The Student enters the user name and password.
- The website verifies the password and sets the user's authorization.
- The Student is given access to the website to perform his tasks.

Alternative Flow: The Student enters invalid username and password then he will not be allowed to enter the website.

Post conditions: The website state is unchanged by this use case.

Use Case Report- Login into the system

Use Case : Edit student details

Goal in context: Edit the details of a student

Brief Description: This use case is used when the student wants to edit the personal details of the himself/herself already existing in the database.

Preconditions:

The Student must be logged into the system in order for this use case to begin.

The details of the student must pre-exist in the database

The student must be enabled by administrator.

Basic Flow:

The Student logs onto the System.

The Student can edit following keys:-

- Name
- Gender
- DOB
- Contact no
- Fathers Name
- Country
- City
- Email
- Address

The Website updates the database according to edited details.

The student details are edited in the database.

Alternative Flow: There is no alternative flow of this use case diagram.

Post conditions: The student details get updated in the database.

5.3 Deployment on Elastic Beanstalk

The AWS Toolkit for Eclipse integrates AWS Elastic Beanstalk management features with your Tomcat development environment to facilitate environment creation, configuration, and code deployment. The toolkit includes support for multiple AWS accounts, managing existing environments, and connecting directly to instances in your environment for troubleshooting.

After Installing the AWS toolkit we have to put Access Key ID & Secret Access Key for Linking to the specific project to access by the user (which we will found on AWS Security credentials)

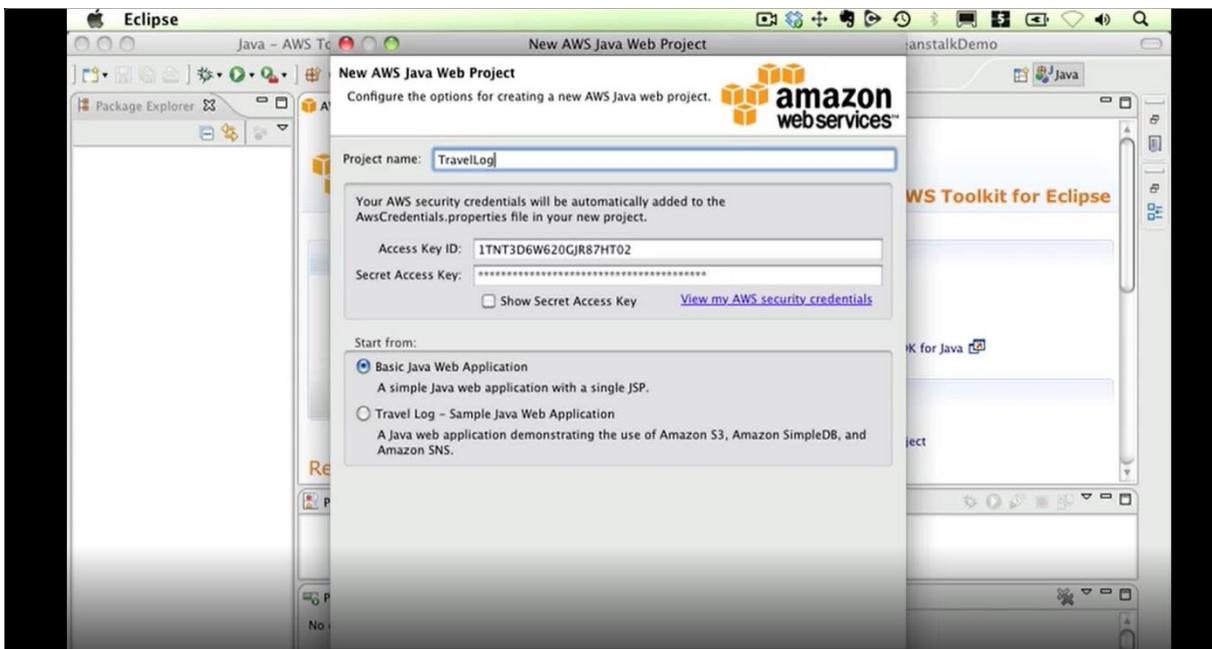
Enabling the access key on cloud for a user through IAM > Users.

The screenshot shows the AWS IAM Management console. On the left, there's a navigation sidebar with 'Identity and Access Management (IAM)' selected. The main area is titled 'Users (1) Info' and contains one user entry:

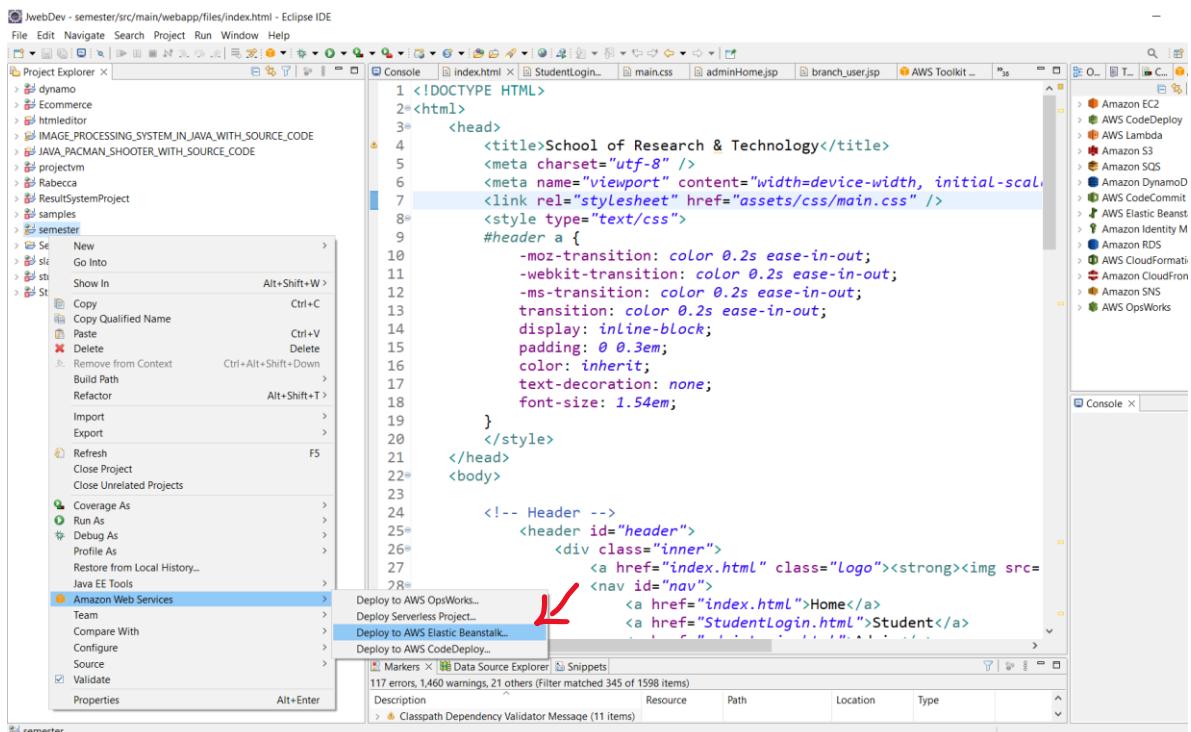
Last activity	MFA	Password a...	Access key...	Access key last used	Console ac...
Yesterday	None	Yesterday	Active - AKIAZFK	Yesterday	Enabled

A red arrow points to the 'Enabled' status column. At the bottom of the page, there are links for 'Feedback', 'Language selection', 'Privacy', 'Terms', and 'Cookie preferences'.

2. Put Access Key ID & Secret Access Key in Eclipse IDE for linking to the specific project to access by the user (which can created through AWS Security credentials) and some basic Configuration.

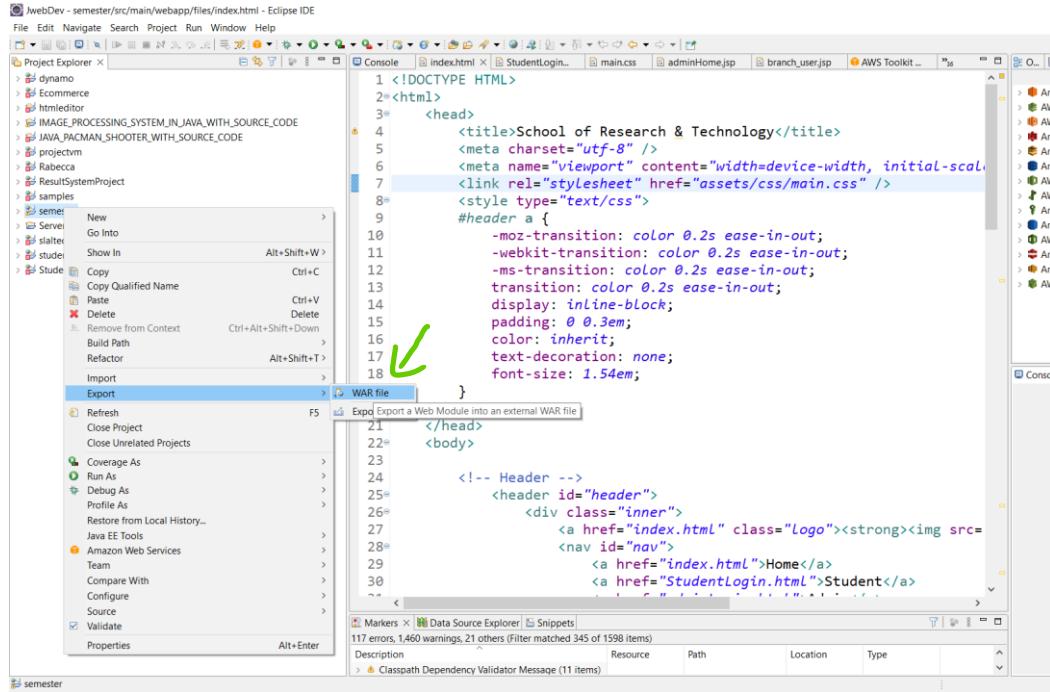


3. Use the Eclipse Java IDE to build and run your application locally before deploying to AWS Elastic Beanstalk.



Now after clicking on Deploy to AWS Elastic Beanstalk , it will run on the local host and becomes deployment ready application.

4. Now Export War file of the Project.



5. Open the AWS Elastic Beanstalk Service on the Cloud and Click on Create a new Application

Application name	Environments	Date created	Last modified	ARN
collegeProj	Collegeproj-env	2023-01-21 21:07:36 UTC+0530	2023-01-21 21:07:36 UTC+0530	arn:aws:elasticbeanstalk:us-east-1:629900772704:application/collegeProj
MajorF	Majorf-env	2023-01-22 15:38:04 UTC+0530	2023-01-22 15:38:04 UTC+0530	arn:aws:elasticbeanstalk:us-east-1:629900772704:application/MajorF
managementsystem	Managementsystem-env Managementsystem-env-1	2023-01-21 19:30:23 UTC+0530	2023-01-21 19:30:23 UTC+0530	arn:aws:elasticbeanstalk:us-east-1:629900772704:application/managementsystem
samplejava	Samplejava-env	2023-01-21 18:51:22 UTC+0530	2023-01-21 18:51:22 UTC+0530	arn:aws:elasticbeanstalk:us-east-1:629900772704:application/samplejava

4-30

After doing some configuration and uploading the WAR file of the application it'll take some minutes for Deployment.

The screenshot shows the AWS Elastic Beanstalk console with a blue banner at the top stating "The new Elastic Beanstalk console experience is now available. We've redesigned the Elastic Beanstalk console to make it easier to use." Below the banner, the navigation bar includes "Elastic Beanstalk" and "Environments". The main content area shows the deployment progress for "Majorf-env". The log output is as follows:

```
3:40pm Created Auto Scaling group policy named:  
amawsautoscalingus-east-1:629900772704:scalingPolicy:294623d8-8ab8-43fd-a257-4af45ec11469:autoScalingGroupName/awseb-e-n9zw6d2rtx-stack-AWSEBAutoScalingGroup-NMREH71GB2G:policyName/awseb-e-n9zw6d2rtx-stack-AWSEBAutoScalingScaleUpPolicy-spZ31B2cT6XA  
3:40pm Created Auto Scaling group named:  
amawsautoscalingus-east-1:629900772704:scalingPolicy:51b0343e-15b5-4e4b-b7c9-f1cd03cb29:autoScalingGroupName/awseb-e-n9zw6d2rtx-stack-AWSEBAutoScalingGroup-NMREH71GB2G:policyName/awseb-e-n9zw6d2rtx-stack-AWSEBAutoScalingScaleDownPolicy-LHWZmNu1gFqQ  
3:40pm Waiting for EC2 instances to launch. This may take a few minutes.  
3:40pm Created Auto Scaling group named:  
awseb-e-n9zw6d2rtx-stack-AWSEBAutoScalingGroup-NMREH71GB2G  
3:48pm Created security group named:  
awseb-e-n9zw6d2rtx-stack-AWSEBSecurityGroup-1XH3VSPZBNXJT  
3:48pm Environment health has transitioned to Pending. Initialization in progress (running for 23 seconds). There are no instances.  
3:48pm Created security group named:  
sg-0e37d0baed4dc9186  
3:48pm Created target group named:  
amawselasticloadbalancing.us-east-1:629900772704:targetgroup/awseb-AWSEB-4WWPTDNL7JE2/3d210865dc5c3739  
3:48pm Using elasticbeanstalk-us-east-1-629900772704 as Amazon S3 storage bucket for environment data.
```

At the bottom, there are links for "Feedback", "Looking for language selection?", "Privacy", "Terms", and "Cookie preferences".

Deployed successfully on AWS Beanstalk

The screenshot shows the AWS Elastic Beanstalk environment details for "Majorf-env". The left sidebar lists "Majorf" and "Majorf-env" sections. The "Majorf-env" section includes options like "Go to environment", "Configuration", "Logs", "Health", "Monitoring", "Alarms", "Managed updates", "Events", and "Tags". The main content area displays the environment summary:

Majorf-env	Majorf-env.eba-xxgnrfsx.us-east-1.elasticbeanstalk.com (e-n9zw6d2rtx)	Application name: Majorf
Health	Running version	Platform
	majorf-source	
Ok	Upload and deploy	Tomcat 8.5 with Corretto 11 running on 64bit Amazon Linux 2/4.3.3 Change
Recent events	Show all	
Time	Type	Details

Got a link of running application of a project running on a cloud.

4-31

Afer Deploying we will link our application to AWS RDS (the database).

Creating the AWS RDS.

The screenshot shows the AWS RDS Management console with the URL <https://us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance:gdb=false;isHermesCreate=true;s3-import=false>. The page title is "Create database". A banner at the top says "We listened to your feedback! Now, create a database with a single click using our pre-built configurations! Or choose your own configurations." Below this, there are two creation methods: "Standard create" (selected) and "Easy create". Under "Engine options", the "MySQL" engine is selected, highlighted with a blue border. Other options include Amazon Aurora, MariaDB, PostgreSQL, Oracle, and Microsoft SQL Server. To the right, a sidebar titled "MySQL" provides information about the MySQL database, including its popularity, supported instance classes (General Purpose, Memory Optimized, and Burstable Performance), automated backup, point-in-time recovery, and replication features. The status bar at the bottom right shows the date as 23-01-2023 and time as 00:45.

Configuring the settings which is sufficient for our Project.

The screenshot shows the AWS RDS Management console with the URL <https://us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance:gdb=false;isHermesCreate=true;s3-import=false>. The page title is "MySQL". On the left, there are sections for "DB instance class" (set to "Standard classes (includes m classes)" with "db.m5d.xlarge" selected), "Storage" (storage type "Provisioned IOPS SSD (io1)", allocated storage "400 GiB", and provisioned IOPS "3000"), and a note about actual IOPS variability. To the right, a sidebar titled "MySQL" provides information about the MySQL database, including its popularity, supported instance classes (General Purpose, Memory Optimized, and Burstable Performance), automated backup, point-in-time recovery, and replication features. The status bar at the bottom right shows the date as 23-01-2023 and time as 00:45.

AWS RDS created.

The screenshot shows the AWS RDS Management Console. The left sidebar has 'Amazon RDS' selected under 'Services'. The main area is titled 'Databases' and displays a table of three database instances. A modal window at the top right provides information about Blue/Green Deployments. The table columns include DB identifier, Role, Engine, Region & AZ, Size, and Status. All instances are listed as 'Available'.

DB identifier	Role	Engine	Region & AZ	Size	Status
awseb-e-byfuef3n86-stack-awsebrdsdatabase-f4kcs1zc0ebc	Instance	MySQL Community	us-east-1a	db.t2.micro	Available
awseb-e-pmbqdszrmz-stack-awsebrdsdatabase-svqfwive7f0u	Instance	MySQL Community	us-east-1f	db.t2.micro	Available
database-2	Instance	MySQL Community	us-east-1c	db.t3.micro	Available

Now we will link our application MySQL database which we using on eclipse ide, to AWS RDS

- Open MySQL Workbench, and then choose the \oplus sign beside MySQL Connections to set up a new connection.
- In the Setup New Connection dialog box, enter a name for your connection.
- In the Parameters section, enter these details:

Host name: Enter the RDS endpoint.

Port: Enter the Port number.

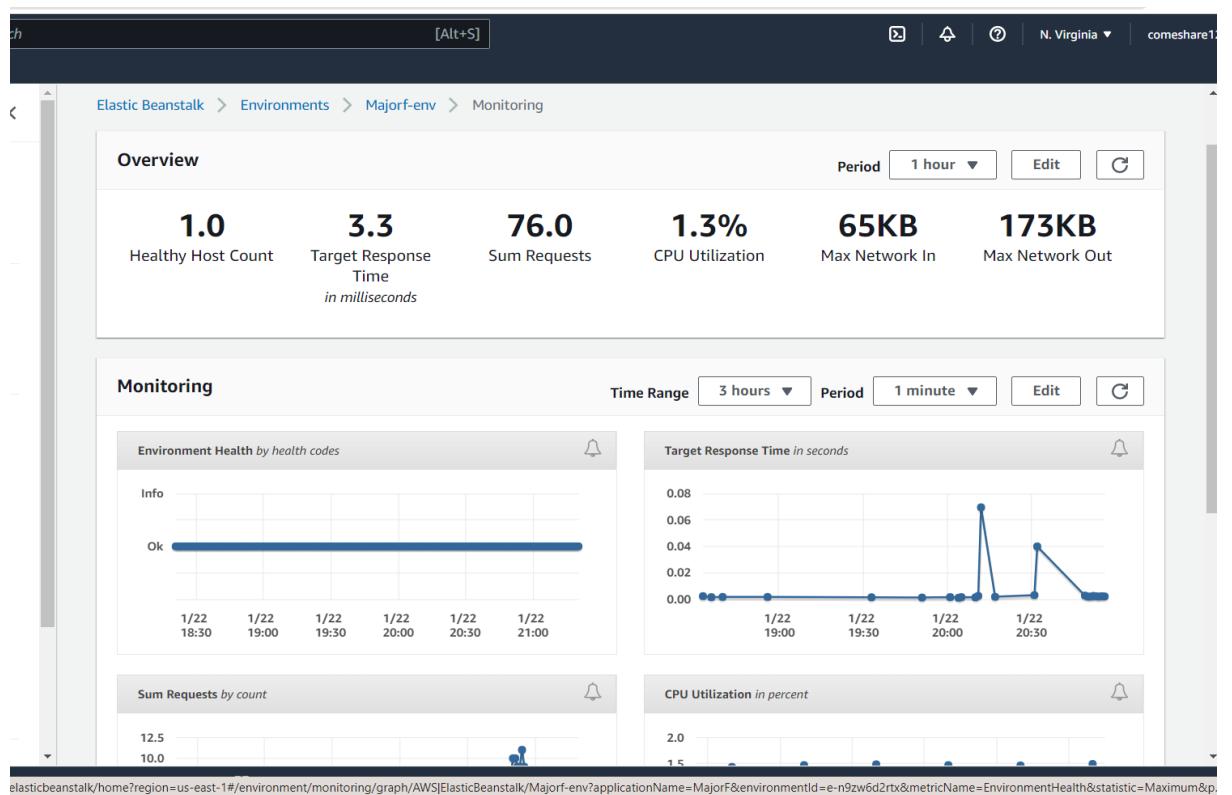
Username: Enter the primary user.

Note: You can get these details from the [Amazon RDS console](#). From the Databases section, choose Instances, and then select the instance that you are connecting to. From the Connectivity and Security tab, choose Configuration. The primary user is listed here.

- Choose Test Connection.

- In the pop-up window that appears, enter the password that you configured when you created the DB instance, and then choose OK.
- After testing your connection, from the Setup new connection dialog box, choose OK to save the connection.

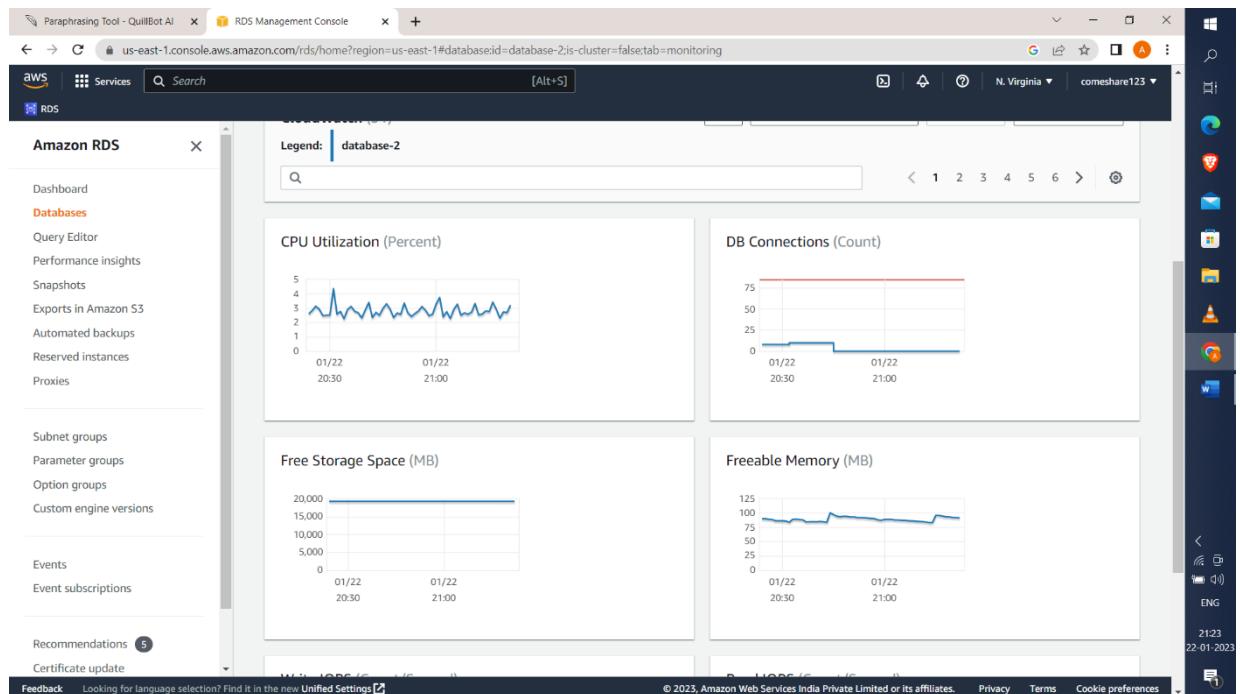
We can monitor the Elastic beanstalk and see various utilizations , such as CPU utilization, Target Response time , sum Requests ,Environment Health and do configuration such as Time range , Period depending upon the various factors

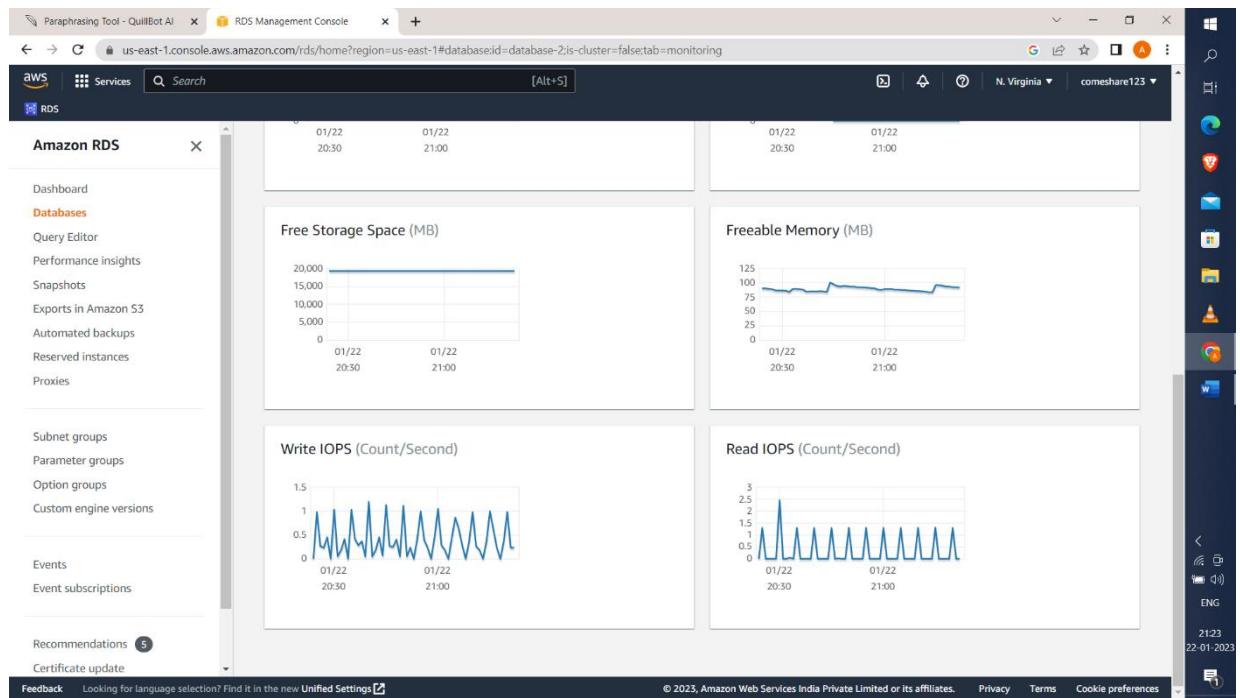


We can get the detailed analysis of Our RDS through the database console

The screenshot shows the Amazon RDS Management Console for a database named 'database-2'. The left sidebar has 'Databases' selected. The main area is titled 'Summary' and displays basic information: DB identifier (database-2), CPU usage (3.22%), Status (Available), Class (db.t3.micro), Role (Instance), Current activity (0 Connections), Engine (MySQL Community), and Region & AZ (us-east-1c). Below the summary are tabs for 'Connectivity & security', 'Monitoring' (which is active), 'Logs & events', 'Configuration', 'Maintenance & backups', and 'Tags'. A 'CloudWatch' section shows 31 metrics, with a chart for 'CPU Utilization (Percent)' and another for 'DB Connections (Count)'. The CPU utilization chart shows values fluctuating between 1% and 5% over time. The DB connections chart shows a sharp drop from approximately 75 to 0 at 20:30 on 01/22.

We can monitor our CPU Utilization , DB Connection ,Storage Space and manymore





4-36

Here we can see all the Databases connected with the Foreign keys or other dependencies.

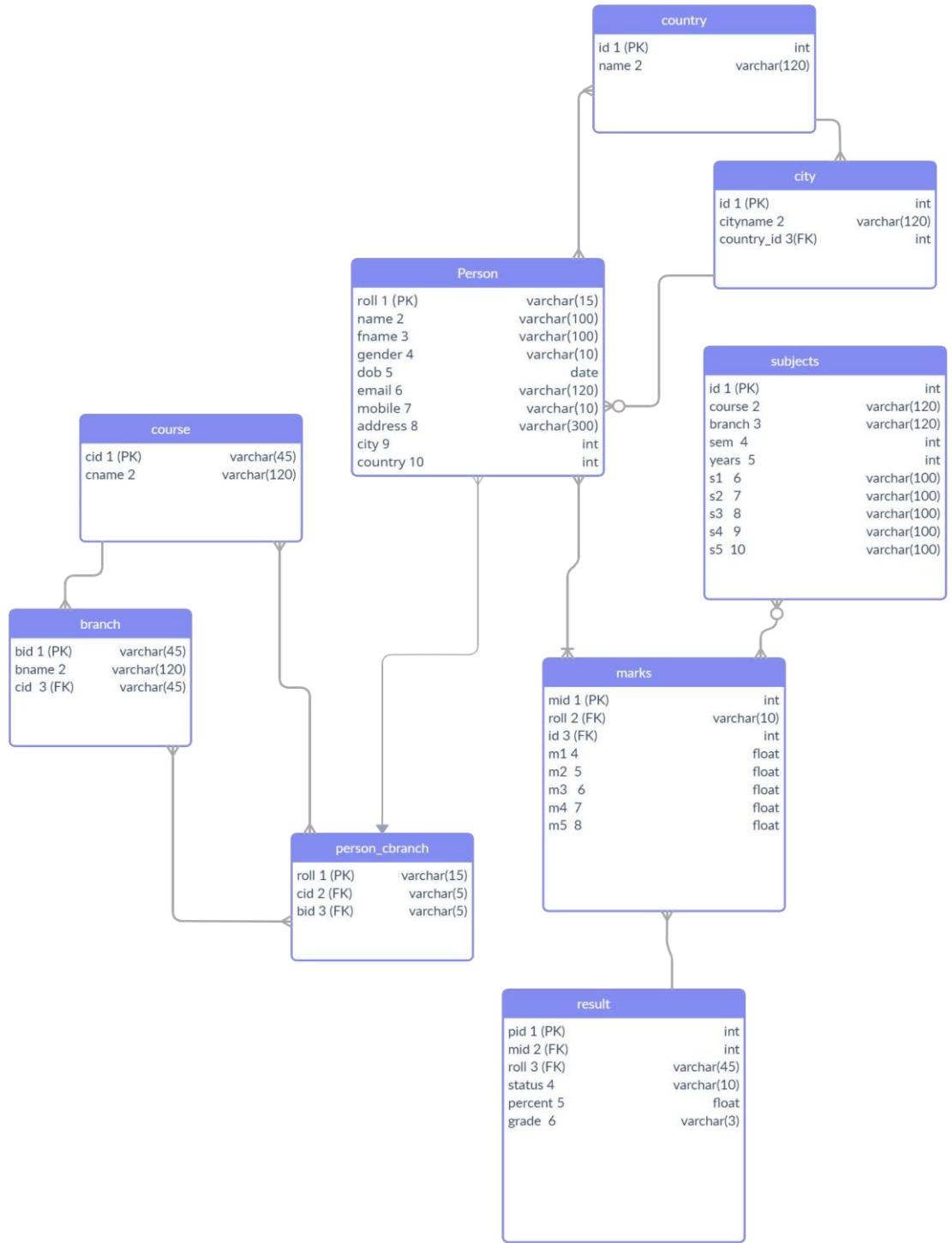


Figure 5.1 Databases

student login	
username 1 (PK)	varchar(45)
pass 2	varchar(35)
Roll 3	varchar(10)

Figure 5.2 DB Student login

1. admin_login:

- username : stores the admin's user id (varchar(45))
- pass : stores the password (varchar(10))

2. student_login:

- username : stores the students's user id (varchar(45))
- pass : stores the password (varchar(10))
- roll : it is a concatenated varchar with int for creating an auto incrementing roll no, everytime a new student registers.(varchar(10))

3. person: this table contains all student data:

- roll : enrollment no. Of the student (varchar(10))
- name: Students's name (varchar(100))

- fname: Fathers's name (varchar(10))
- Gender: (varchar(10))
- DOB : (date)
- Contact no :(varchar(10))
- Country: country id from country table(int)(foreign key)
- City : city id from city table (int)(foreign key)
- Email: (varchar(120))
- Address : (varchar(300))

4. person_cbranch:

- roll: enrollment no. Of the student (varchar(10))
- bid: branch id from branch table (int)(foreign key)
- cid: course id from course table (int)(foreign key)

5. country:

- id: (int)
- name: (varchar (120))

6. city :

- id: (int)
- cityname: (varchar (120)) • country_id: (int)(foreign key)

7. course:

- cid: int (primary key) •
cname: course
name(varchar(120))

8. branch :

- bid:int (primary key)
- bname: course name(varchar(120))
- cid: course id .int (foreign key)

9. subjects:

- id: int (primary key)
- course: course name(varchar(120))
- branch: branch name(varchar(120))
- sem: int
- year:int
- s1: subject 1 varchar(100)
- s2: subject 2 varchar(100)
- s3: subject 3 varchar(100)
- s4: subject 4 varchar(100)
- s5: subject 5 varchar(100)

10. marks:

- mid: id int (Primary key)
- roll: varchar(10)
- id: int (Foreign key)
- m1: marks 1 (float)
- m2: marks 2 (float)

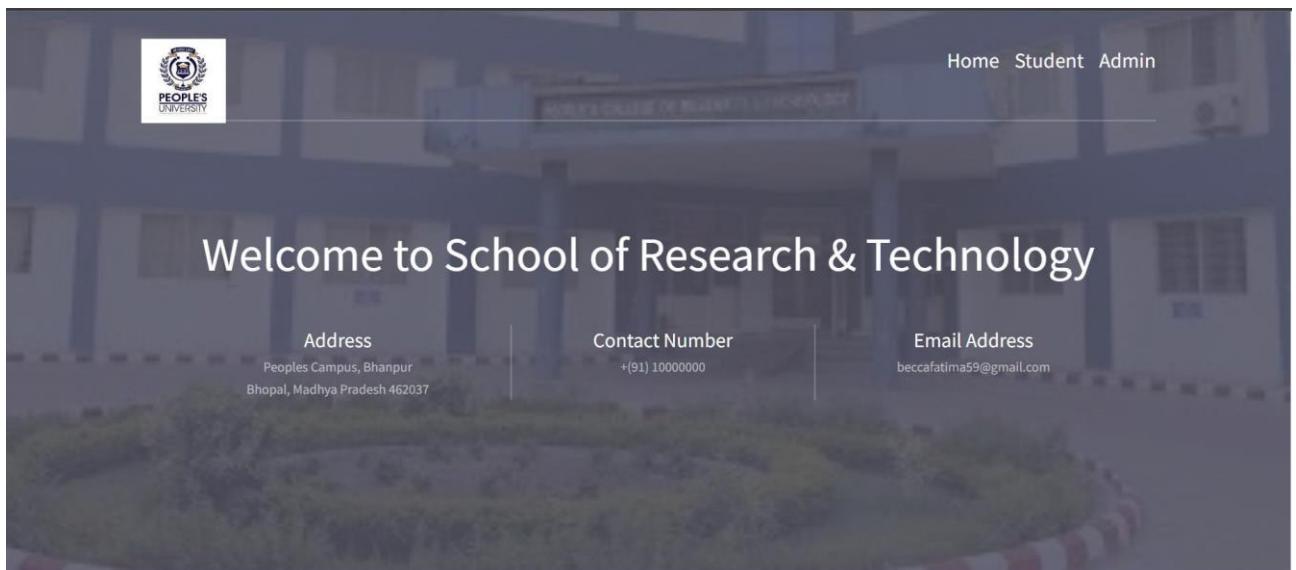
- m3: marks 3 (float)
- m4: marks 4 (float)
- m5: marks 5 (float)

11. result:

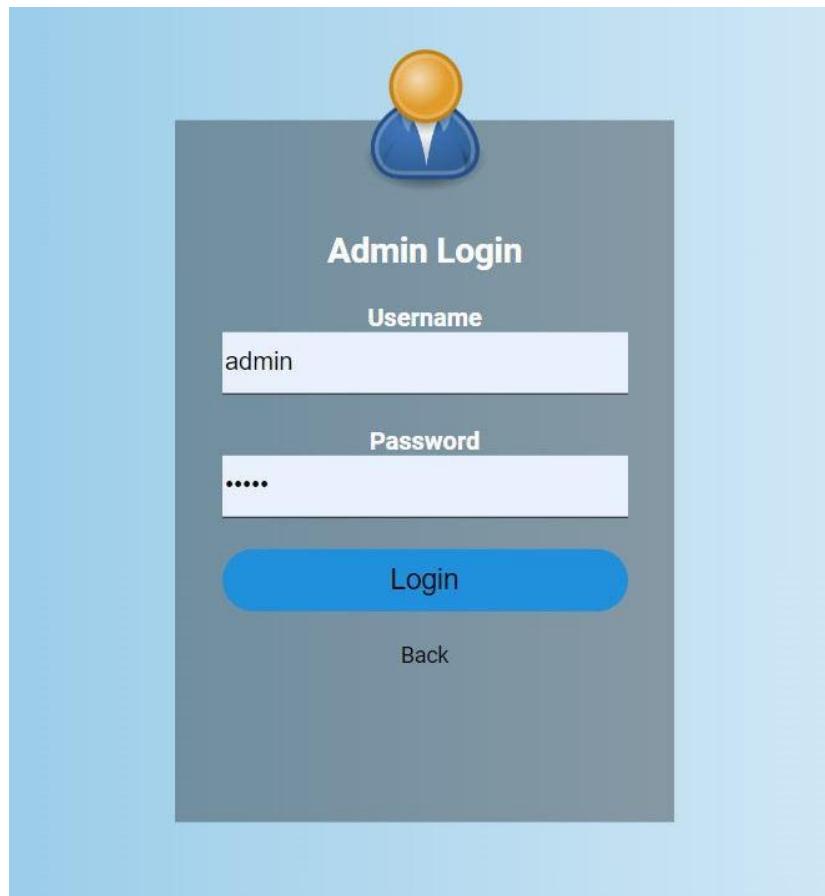
- pid: id int (Primary key)
- mid: int (Foreign key)
- roll: varchar(10) (Foreign key)
- status: varchar(5)(eg. pass/fail)
- percent: float
- grade: varchar(3)

5.4 SNAPSHOT

Index page



Admin Login



After Login -> Add New Student

Add New Student Add New Course Add New Branch Add New Subject Insert New Result
Registered Students All Student Result Logout welcome :admin

Add New Students

Course:

Branch Name:

Roll Number:

Name:

Father's Name:

Gender: Male Female Trans

D.O.B.: dd-mm-yyyy

Email:

Mobile:

Address:

Country:

City:

Save

Note: Any errors occur then contact Rabecca Fatima. Designed & Developed by Rabecca Fatima
All Right Reserved @Rabecca Fatima :: 2022-2025

when save clicked

The screenshot shows a form with two dropdown menus: 'Country' set to 'India' and 'City' set to 'Meerut'. Below the form is a blue button labeled 'Save'. Underneath the button, a message reads 'Data Saved :Details of Abdul Hamid are saved.'

Country: India

City: Meerut

Save

Data Saved :Details of Abdul Hamid are saved.

Add New Course

The screenshot shows a form titled 'Course' with two input fields: 'Course ID' and 'Course Name'. Below the form is a blue button labeled 'Save'. The top navigation bar includes links for 'Add New Course', 'Add New Branch', 'Insert New Result', 'Add New Subject', 'Registered Students', 'All Student Result', 'Logout', and 'welcome :admin'.

Add New Course Add New Branch Insert New Result Add New Subject Registered Students All Student Result Logout welcome :admin

Course

Course ID

Course Name

Save

Add New Branch

Branch

Course	Bachelor in Business M ▾
Branch ID	Bachelor in Business Management Bachelor in Commerce Bachelor in Computer Application Bachelor in Science Btech Masters in Business Management Mtech
Branch Name	

branch saved

Branch

Course	Btech
Branch ID	CSE-CCV
Branch Name	Computer science engineering with specialization in cloud comp

Save

Data Saved : Branch: Computer science engineering with specialization in cloud computing saved.

Add New Subjects according to course , branch ,semester ,year

Add New Subject

Course	Btech
Branch Name	Computer science engineering with specialization in cloud co
Year	1
Semester	1
Subject 1	Open Source & Standards
Subject 2	C Programming Language
Subject 3	Maths-I
Subject 4	Communication skills
Subject 5	Engineering Chemistry

Save

Data Saved :Subjects are saved.

Here we will be adding marks of the students , by selecting course,branch,semester,year

Insert New Result

Roll Number	PU-005
Course	Btech
Branch	Computer science engineering with specialization in cloud cc
Semester	1
Year	1

Show Subjects Save

After clicking on show subjects add the marks and save

Insert New Result

Roll Number	PU-005
Course	Btech
Branch	Computer science engineering
Semester	1
Year	1
C Programming	78.8
Maths-1	67.89
Chemistry	49
Open Source & Standards	
Communication skills	
Show Subjects Save	

Display all registered students

Course	Branch Name	Roll Number	Show							
			Show							
ROLL NO.	NAME	DOB	BRANCH NAME	COURSE NAME	GENDER	FATHER NAME	CONTACT	EMAIL	ADDRESS	MODIFY
PU-001	RABECCA	2001-01-02	CSE	BTI	FEMALE	ARSHAD	9834567890	RABECCA@123.COM	88/121,PREM NAGAR	EDIT
PU-002	ROHAN KUMAR	2001-06-05	CSE	BTI	MALE	RAJAT KUMAR	1234567890	RAJAT@123.COM	99/121-JAHANGIRABAD,BHOPAL	EDIT
PU-003	AASIM AHSAN	2000-12-04	CSE	BTI	MALE	AHSANUL HODA	9087907690	AASIMAHSANC7@GMAIL.COM	77/80,KIRANA ROAD,BHIIHAR	EDIT
PU-004	KHUSHBU JINDAL	2001-01-13	BCN	BCOM	FEMALE	RAMESH JINDAL	9638209182	KH5KH123@GMAIL.COM	88/109-1,ANSOL, WEST BENGAL	EDIT

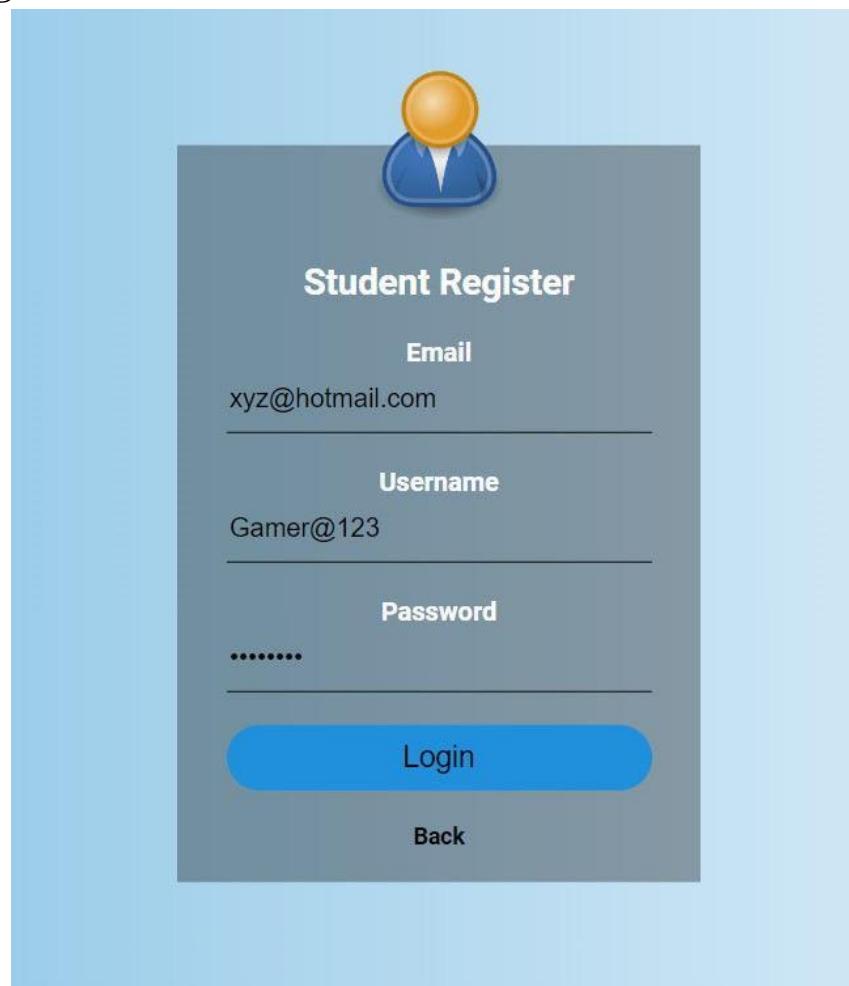
Using course,branch,roll number as filters, one can easily view the data

Display result of all students:

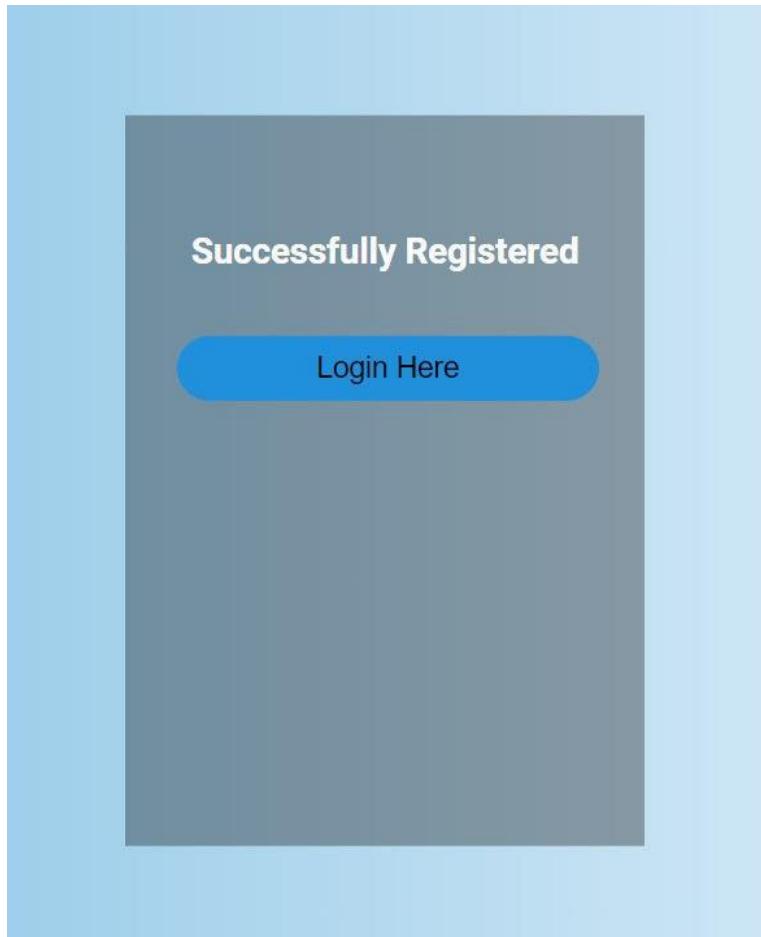
Using course,branch,roll number as filters, one can easily view the result of a particular student

Add New Student	Add New Course	Add New Branch	Add New Subject	Insert New Result	Registered Students	All Student Result	Logout	welcome :admin						
Course	Btech													
Branch Name	Computer science engineering													
Roll Number	PU-001													
Show														
ROLL NO.	SUBJECT1	MARKS1	SUBJECT2	MARKS2	SUBJECT3	MARKS3	SUBJECT4	MARKS4	SUBJECT5	MARKS5	PERCENT	GRADE	STATUS	ACTION
PU-001	C PROGRAMMING ⁹⁹		MATHS-1	99	CHEMISTRY	99	OPEN SOURCE & STANDARDS	99	COMMUNICATION SKILLS ⁹⁹	99		A+	PASS	EDIT

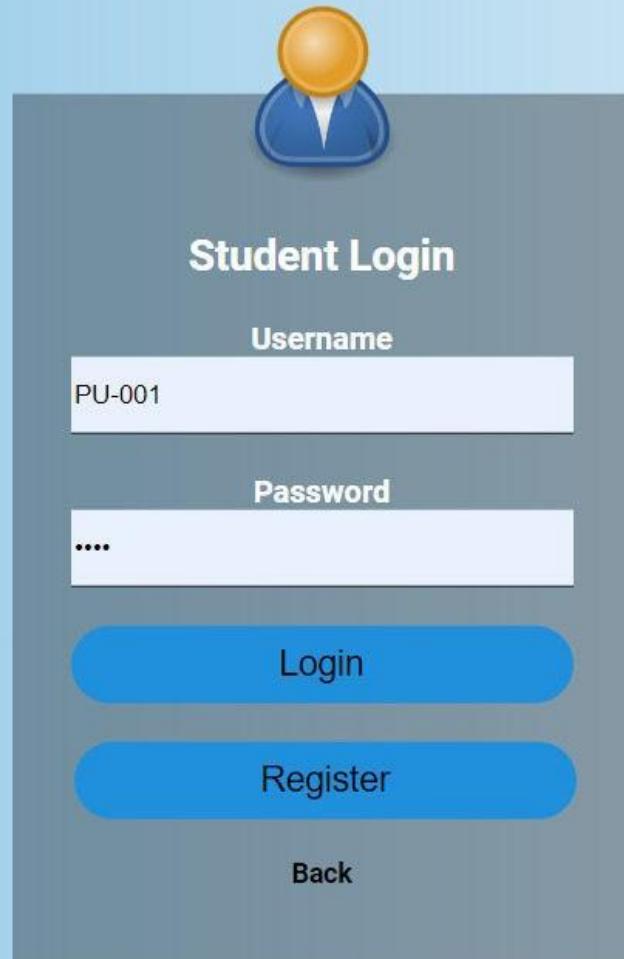
Student Register



Successfully registered



Student Login



Student Home page

The screenshot shows a web-based student management system. At the top, there is a navigation bar with links for "Modify Details", "View Result", "Logout", and a welcome message "welcome :PU-001". Below the navigation bar is a section titled "Student Details" which displays various personal and academic information.

Student Details	
Course :	Btech
Branch :	Computer science engineering
Roll Number :	PU-001
Name :	Rabeeca
Father's Name :	Arshad
Gender :	Female
D.O.B :	2001-01-02
Email :	rabeeca@123.com
Mobile :	9834567890
Address :	88/121, PREM NAGAR
Country :	India

View Result

PEOPLES UNIVERSITY

Modify Details View Result Logout welcome :PU-001

Select Year & Semester

Year: 1 Semester: 1 Show

PEOPLES UNIVERSITY

Modify Details View Result Logout welcome :PU-001

INSTITUTION NAME: PEOPLES UNIVERSITY	COURSE NAME: BTECH	BRANCH NAME: COMPUTER SCIENCE ENGINEERING	YEAR: 1	SEMESTER: 1	ROLLNO: PU-001
NAME: RABECCA	FATHER NAME: ARSHAD	GENDER: FEMALE			

Course	Type	Full Marks	Passing Marks	Obtained Marks	
Name					
C Programming	Theory	100	30	99.0	
Maths-1	Theory	100	30	99.0	
Chemistry	Theory	100	30	99.0	
Open Source & Standards	Theory	100	30	99.0	
Communication skills	Theory	100	30	99.0	
	Total Marks	500	180	495.0	
Percentage	99%	Grade	A+	Status	pass

Update student details

Modify Details

Course: BTI

Branch Name: CSE

Roll Number: PU-001

Name: Rabecca

Father's Name: Arshad

Gender: Male Female Trans

D.O.B: 02-01-2001

Email: rabecca@123.com

Mobile: 9834567890

Address: 88/121,PREM NAGAR

Country: India

City: Indore

Update

Note: Any errors occur then contact Rabecca Fatima. Designed & Developed by Rabecca Fatima

All Right Reserved @Rabecca Fatima :: 2022-2025

Update Student result if any(only done by the admin)

Update Result

Roll Number	PU-002
Course	Btech
Branch	Computer science engineering
Semester	1
Year	1

[Show Subjects](#)

C Programming	99.0
Maths-1	72.99
Chemistry	60.0
Open Source & Standards	45.0
Communication skills	56.0

[Update](#)

Data SavedDetails of PU-002 are updated.

5.5 FLOWCHART

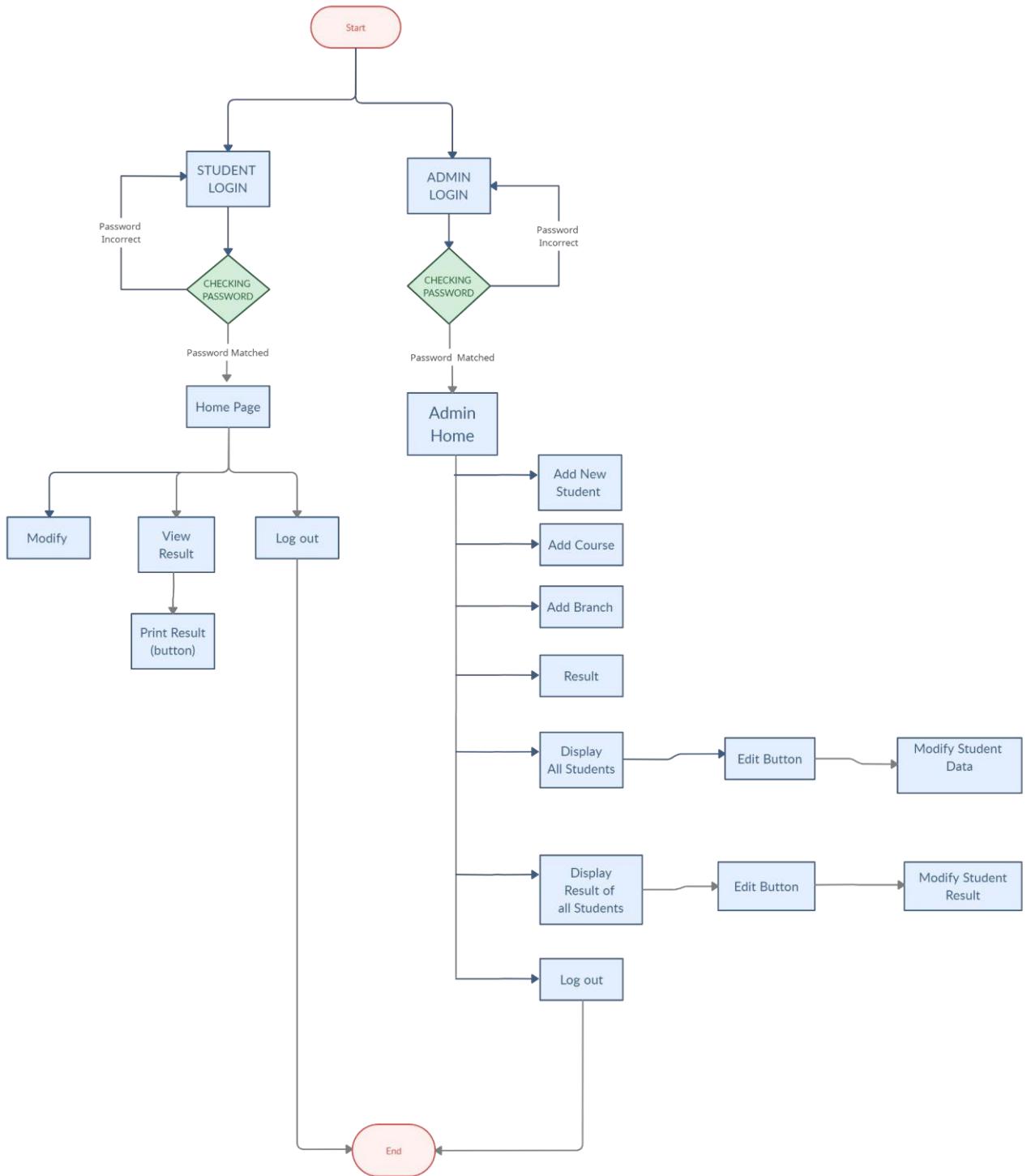


Figure 5.3 Project Flowchart

SCOPE OF THE PROJECT

- The Student Management System(SMS) can be enhanced to include some other functionality like attendance management. □
- Talent management of students based on their performance evaluation can be added. □
- Social networking can also be added wherein students can interact with each other. □
- Online class functionality can be added. □
- Can evolve as an online institution. □
- Functionality of chat and messages can be added. □
- Online exam functionality can be added. □
- Online resume builder functionality can also be added.

Bibliography

- www.google.com
- www.w3school.com
- JavaTpoint
- Jdbc, Servlets, And Jsp Black Book, New Edition (With Cd)