

B. Tech. project Grace Marks Allocator– G 22

PROJECT REPORT-I

Submitted by

S. No	Name	Roll Number
1.	G Lokesh	CB.EN.U4CSE18117
2.	P Pruthve Rej	CB.EN.U4CSE18136
3.	N Pavan Kruthik	CB.EN.U4CSE18137
4.	P Upendra	CB.EN.U4CSE18138
5.	Sk Awez	CB.EN.U4CSE18148

is fulfilment of the requirements for the Course – 15CSE376 (Net Centric Programming)

**BACHELOR OF TECHNOLOGY IN
COMPUTER SCIENCE AND ENGINEERING**



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING**

AMRITA SCHOOL OF ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE -641112

August 2021

Table of Contents

S.No	Section Name	Page Number
1	Abstract	3
2	Introduction to web designing	4
3	Design Diagram	6
4	Implementation	7
5	Evaluation sheet	25

Abstract

- Every year the tracking of paper publications that provide students with grace marks and finding the right subject to which the marks can be added in order to increase the grade of the student in the best way possible is a very manually hectic job
- We have designed a software that eases the tedious process of handling the allocation of Grace marks to students belonging to a semester, impartially
- To do so, we set rules that govern the generation of grace marks and constrain them. Grace marks are generated on the basis of the proofs that have been submitted and verified
- The generated marks are then added to the existing marks and published.

Introduction to web designing:

What is HTML?

HTML is the language for describing the structure of Web pages. HTML gives authors the means to:

- Publish online documents with headings, text, tables, lists, photos, etc.
- Retrieve online information via hypertext links, at the click of a button.
- Design forms for conducting transactions with remote services, for use in searching for information, making reservations, ordering products, etc.
- Include spread-sheets, video clips, sound clips, and other applications directly in their documents

With HTML, authors describe the structure of pages using markup. The elements of the language label pieces of content such as “paragraph,” “list,” “table,” and so on

CSS:

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML. CSS describes how elements should be rendered on screen, on paper, in speech, or on other media. CSS is used to style and layout web pages — for example, to alter the font, colour, size, and spacing of your content, split it into multiple columns, or add animations and other decorative features. This module provides a gentle beginning to your path towards CSS mastery with the basics of how it works, what the syntax looks like, and how you can start using it to add styling to HTML.

JavaScript:

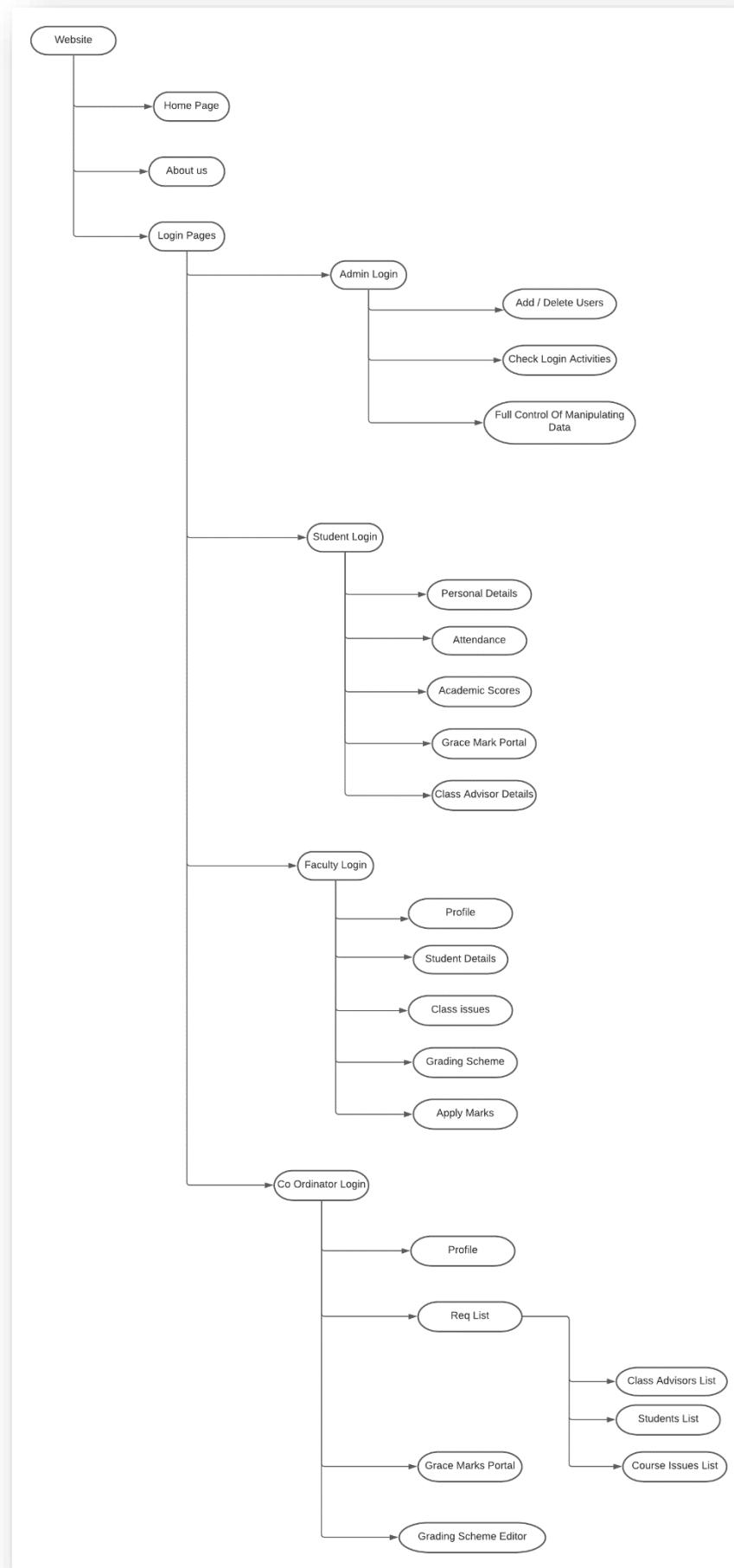
JavaScript (JS) is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat. JavaScript is a prototype-based, multi-paradigm, single-threaded, dynamic language, supporting object-oriented, imperative, and declarative styles.

Embedded JavaScript:

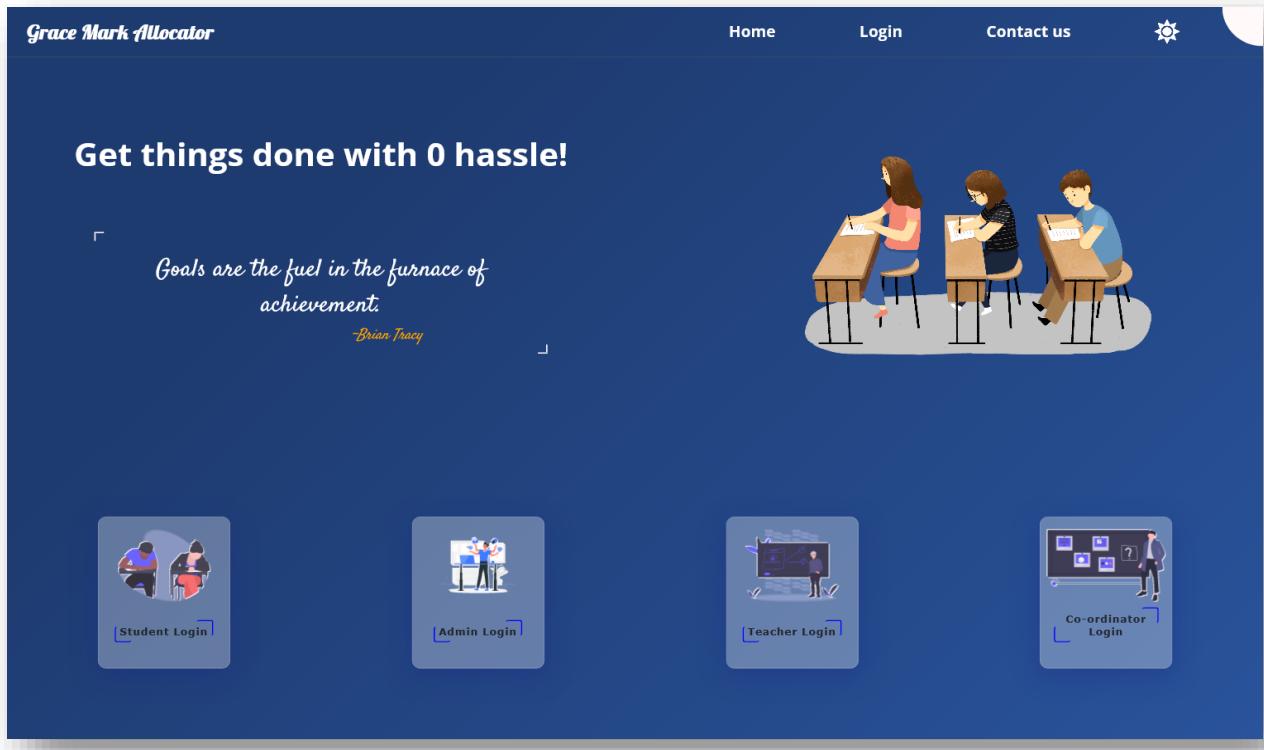
You can embed JavaScript in an HTML document in the following ways: As statements and functions within a <SCRIPT> tag. ... See "Specifying a file of JavaScript code". By specifying a JavaScript expression as the value for an HTML attribute. See "Using JavaScript expressions as HTML attribute values".



Design diagram:



Implementation:



Description:

This page is the landing page of our website where users can choose their respective login page.

Code snippet:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
    <link rel="stylesheet" href="http://localhost:5000/css/home.css" />
  </head>
  <form method="GET", action="http://localhost:8000/">
    <body>
      <div class="header">
        <h2 class="head">Grace Mark Allocator</h2>
```

```
<div class="login">
    <a href="">Home</a>
    <a href="#box">Login</a>
    <a class="contact" href="">Contact us</a>
    
    <!-- <a class="mode" href="">Mode</a> -->
</div>
</div>
<div class="container">
    <div class="mini-cont">
        <h1>Get things done with 0 hassle!</h1>

        <div class="quotes">
            <blockquote class="quote"></blockquote>
            <div class="author"></div>
        </div>
    </div>
    <div class="image">
        
    </div>
</div>
<section id="section1">
    <div class="box" id="box">
        
        <div class="btn-box">
            <button formaction="/Student">Student Login</button>
        </div>
    </div>
    <div class="box admin">
        
        <div class="btn-box">
            <button formaction="/Admin">Admin Login</button>
        </div>
    </div>
    <div class="box">
        
        <div class="btn-box">
            <button formaction="/Faculty">Teacher Login</button>
        </div>
    </div>
    <div class="box">
```

```

<div class="btn-box">
    <button formaction="/Coordinator">Co-ordinator Login</button>
</div>
</div>
</section>
<nav class="nav-bar">
    <ul class="nav-links">
        <div class="ourTeam">
            <h3><a href="http://localhost:5000/contact.html"> Our Team </a></h3>
            
        </div>
        <div class="git">
            <h3>
                <a href="https://github.com/awezrebel/Grace-marks-Calculator"
                    target="blank"
                >Github</a>
            </h3>
            
        </div>
    </ul>
    <div class="contacts">
        <h2>Stay in touch!</h2>
    </div>
</nav>

<script src="http://localhost:5000/js/home.js"></script>
</body>
</html>
```

Welcome Student



Sign in

Username _____

Password _____

submit

[Forgot password ?](#)

Welcome Admin



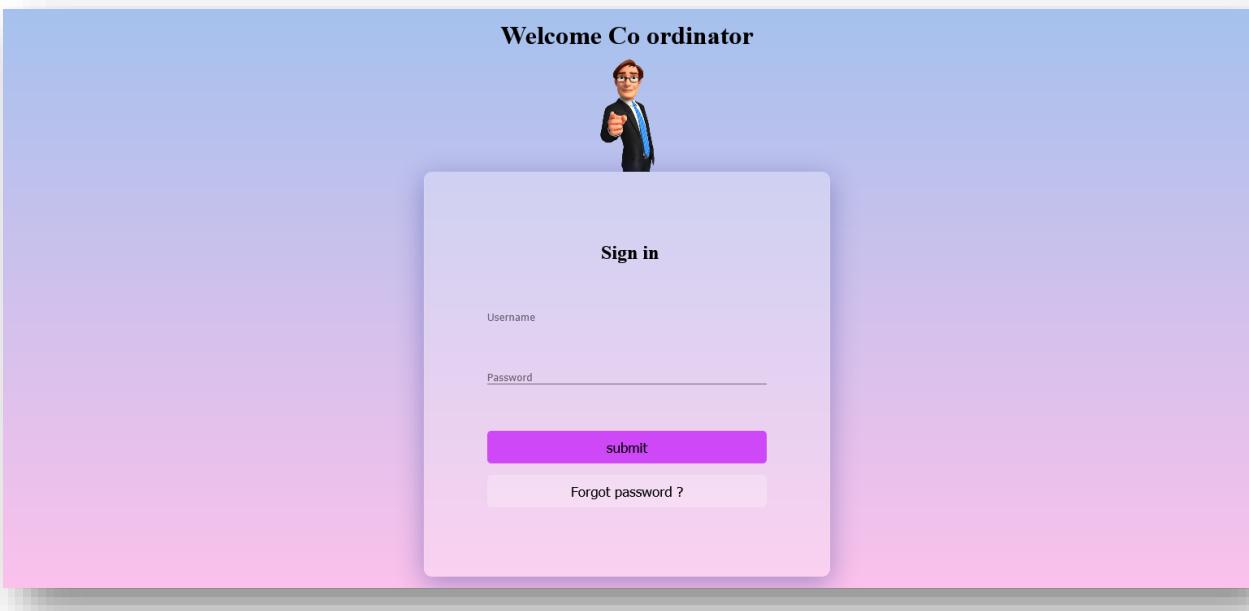
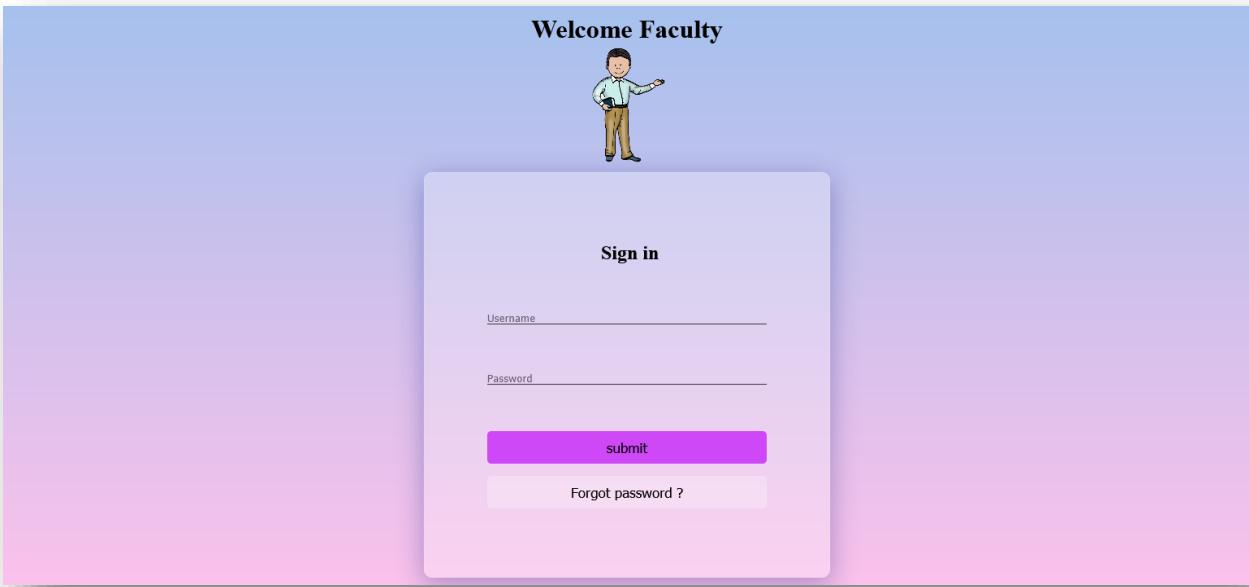
Sign in

Username _____

Password _____

submit

[Forgot password ?](#)



Description:

These are the different login pages done using single ejs file

Code snippet:

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
    <link rel="stylesheet" type="text/css" href="http://localhost:5000/css/style.css" />
</head>
<style>
    a{
        color: black;
        font-size: larger;
        text-align: center;
        margin-top: 5%;
        text-decoration: none;
    }
</style>

<form method="GET", action="http://localhost:8000/">

<body>

    <div class="main-container">
        <div class="slider-container">
            <div class="left-slide">
                <div class="sign-in">
                    <h1>
                        <%= "Welcome " + welcome.Name %>
                    </h1>
                    <img src ="<%=welcome.image %>" width="100px">

                    <div class="sign-in-box">

                        <div class="head">

                            <h2>Sign in</h2>
                        </div>
                        <div class="username-field">
                            <i></i>
                            <input type="text" placeholder="Username" name="username" />
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</body>

```

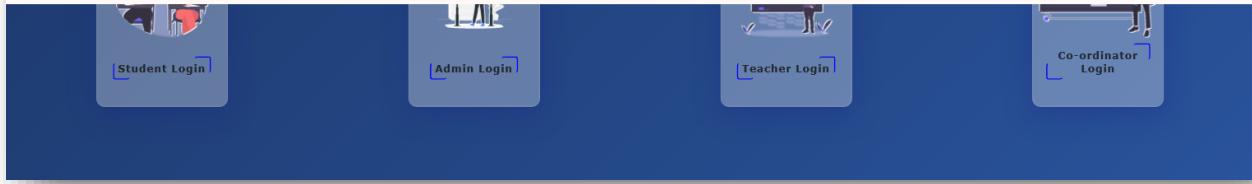
```
        </div>
        <div class="password-field">
            <i></i>
        <input type="password" placeholder="Password" name="password"/>
        </div>
        <button formaction="/homelogin">submit</button>
        <br><br>

<button style="backgroundcolor:inherit;" formaction="forgot">Forgot password ?</button>
        </div>
        </div>
    </div>
</body>

</html>
```

Our Team 
Github 

Stay in touch!



Meet the creators!

[Home](#) [About Us](#)

Shaik Awez
Pavan Kruthik.N
Upendra.P
Pruthve Rej
Lokesh Gupta



Back end Developer

6303731463

shaikawez@gmail.com

Description:

About us and git hub link From Home page

Code snippet:

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Contact Us</title>
    <link rel="stylesheet" href="http://localhost:5000/css/contact.css" />
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/fontawesome/5.15.4/css/all.min.css" integrity="sha512-1ycn6IcaQQ40/MKBW2W4Rhis/DbILU74C1vSrLJxCq57o941Ym01SwNs0MqvEBFlcgUa6xLiPY/NS5R+E6ztJQ==" crossorigin="anonymous" referrerPolicy="no-referrer" />

</head>

<style>
.home {
position: absolute;
top: 0.8rem;
right: 10rem;
}

.home button {
background: #ed710c;
background-image: -webkit-linear-gradient(top, #ed710c, #e87c2a);

background-image: linear-gradient(to bottom, #ed710c, #e87c2a);

border-radius: 4px;
text-align: center;

color: #ffffff;
font-size: 18px;
padding: 0.5rem;
padding-right: 1rem;
padding-left: 1rem;
text-decoration: none;
}
```

```
border: none;
cursor: pointer;
transition: all 0.5s ease-in-out;
}

.home button:hover {
background: #3cb0fd;
text-decoration: none;
}


```

</style>

```
<body>
<div class="header">
    <div class="messsage">
        <h1>Meet the creators!</h1>
    </div>
    <div class="about-us">
        <button>About Us</button>

    </div>
    <div class="home">
        <form method="get" action="http://localhost:8000/home">
            <button type="submit">Home</button>
        </form>

    </div>
</div>

<div class="img-container">
    <div class="image">
        
    </div>

    <div class="creator-btns">
        <div class="click-btns">
            <a href="#item1">

                <button>Shaik Awez</button>
            </a>
        </div>
        <div class="click-btns">
            <a href="#item2">
```

```
        <button>Pavan Kruthik.N</button>
    </a>

</div>
<div class="click-btns">
    <a href="#item3">

        <button>Upendra.P</button>
    </a>

</div>
<div class="click-btns">
    <a href="#item4">

        <button>Pruthve Rej</button>
    </a>

</div>
<div class="click-btns">
    <a href="#item5">

        <button>Lokesh Gupta</button>
    </a>

</div>
</div>

<div class="creator-container">
    <div class="item" id="item1">
        
        <span>Back end Developer</span>
        <div class="details">
            <i class="fas fa-phone-square-alt"></i>
            <span>6303731463</span>
            <div class="mail">

                <i class="fas fa-envelope"></i>
                <span>shaikawez@gmail.com</span>
            </div>
        </div>
    </div>

    </div>
```

```
<div class="item" id="item2">
    
    <span>Front end Developer</span>
    <div class="details">
        <i class="fas fa-phone-square-alt"></i>
        <span>8341900899</span>
        <div class="mail">

            <i class="fas fa-envelope"></i>
            <span>kruthik24@gmail.com</span>
        </div>
    </div>

</div>
<div class="item" id="item3">
    
    <span>Front end Developer</span>
    <div class="details">
        <i class="fas fa-phone-square-alt"></i>
        <span>9182410276</span>
        <div class="mail">

            <i class="fas fa-envelope"></i>
            <span>p_upendra@gmail.com</span>
        </div>
    </div>
</div>
<div class="item" id="item4">
    
    <span>Tester</span>
    <div class="details">
        <i class="fas fa-phone-square-alt"></i>
        <span>9381044045</span>
        <div class="mail">

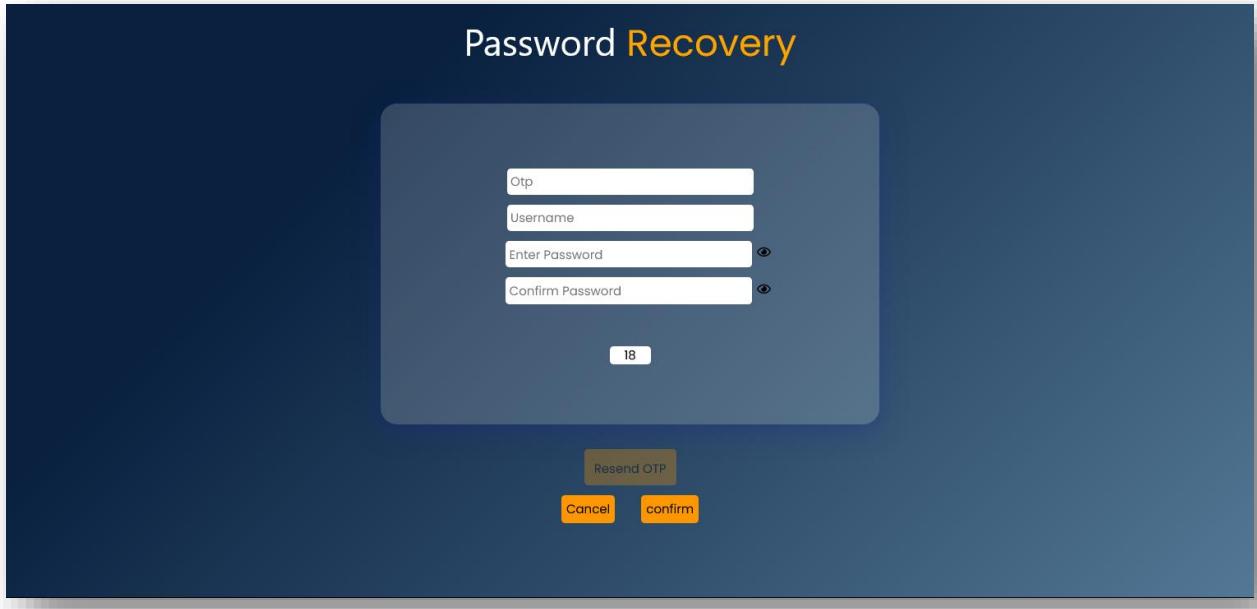
            <i class="fas fa-envelope"></i>
            <span>p_pruthve@gmail.com</span>
        </div>
    </div>
</div>
<div class="item" id="item5">
    
    <span>Tester</span>
    <div class="details">
```

```
<i class="fas fa-phone-square-alt"></i>
<span>8142838171</span>
<div class="mail">
    <i class="fas fa-envelope"></i>

    <span>g_lokesh@gmail.com</span>
</div>
</div>
</div>

<script src="http://localhost:5000/js/contact.js"></script>
</body>

</html>
```



Description:

Forgot password page , with the help of otp generated to mobile can reset password , done validations for each input feild

Code snippet:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <title>OTP</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.13.0/css/all.min.css">

<body>

  <header>
    <h2>Password <span>Recovery</span></h2>
  </header>
  <form method="GET", action="http://localhost:8000/">

    <div class="inputs">

      <input type="text" placeholder="Otp" name="otp">
      <input type="text" placeholder="Username" name="uname1">
      <div class="password">
        <input type="password" placeholder="Enter Password" name="pwd" id="pwd">
        <em class="far fa-eye" onclick="password()"></em>
      </div>
      <div class="confirm-password">

        <input type="password" placeholder="Confirm Password" name="cpwd" id="cpwd">
        <em class="far fa-eye" onclick="confirmPassword()"></em>
      </div>
      <div id="countdown"></div>

    </div>

  </form>

</body>
```

```

<script>
    function password() {
        var x = document.getElementById("pwd");
        if (x.type === "password") {
            x.type = "text";
        } else {
            x.type = "password";
        }
    }

    function confirmpassword() {
        var x = document.getElementById("cpwd");
        if (x.type === "password") {
            x.type = "text";
        } else {
            x.type = "password";
        }
    }
</script>

<button class="w3-btn w3-orange" id="mybutton" formaction="/otp">Resend OTP</button> <br><br>
<p>

<button formaction="/home" , class="w3-btn w3-orange">Cancel</button> &ampnbsp &ampnbsp
<button type="submit" , class="w3-btn w3-orange">confirm</button> <br><br>
<p></p>
<script>
    const htmlBtn = document.getElementById("mybutton");
    htmlBtn.disabled = true;
</script>

<script>
    alert("otp sent successfully")
    var timeleft = 20;
    var downloadTimer = setInterval(function () {
        if (timeleft <= 0) {
            clearInterval(downloadTimer);
            document.getElementById("countdown").innerHTML = " ", htmlBtn.disabled = false;
        } else {
            document.getElementById("countdown").innerHTML = timeleft;
        }
    }, 1000);
</script>

```

```
        }
        timeleft -= 1;
    }, 1000);

</script>

</form>

</body>

</html>
```

Updated Successfully



Logout successful



You have logged out successfully

For security reasons, exit your web browser

Description :

Basic logout page (Backward prevention script added), updation Acknowledge page after password reset

Code snippet:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
body{
    background-image: linear-gradient(to top, #fbc2eb 0%, #a6c1ee 100%);
    height: 95.5vh;

    justify-content: center;
    align-items: center;
    text-align: center;
}
img {
    border-radius: 50%;
}
</style>
<title>Logout</title>
<form method="GET", action="http://localhost:8000/">
</head>
<body>

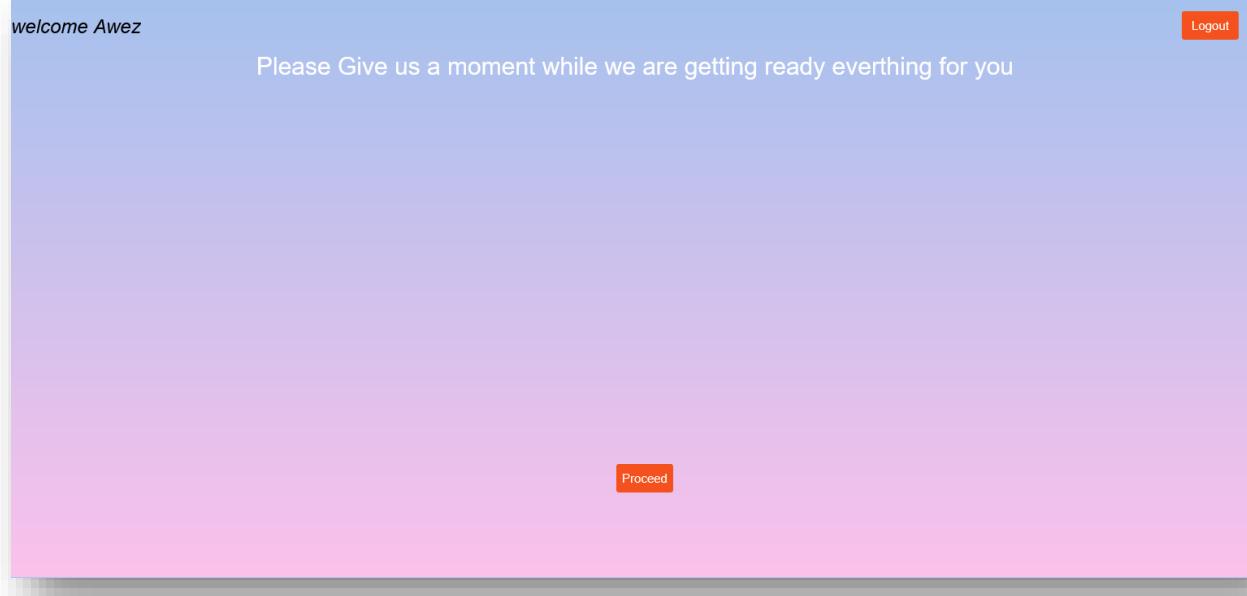
<script type="text/javascript">
    window.history.forward();
    function noBack()
    {

        window.onbeforeunload = function() { return "Your work will be lost."
; };
    }
</script>

<body onLoad="noBack();" onpageshow="if (event.persisted) noBack();" onUnload="">
```

```
<h1>Logout successful </h1>
</img><br><br>
<p1> You have logged out successfully </p1><br>
<br>
<p1> For security reasons, exit your web browser </p1> <br> <br>

<script type = "text/javascript" >
function preventBack() { window.history.forward(); }
setTimeout("preventBack()", 0);
window.onunload = function () { null };
</script>
</body>
</html>
```



Description:

Welcome page with name , Fetched According to Login id with an Loading Animation

Code snippet:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <link rel="stylesheet" href="css/welcome.css">

<body>
<form method="GET", action="http://localhost:8000/">

    <% loginwelcome.forEach(function (login) { %>

        <h1>
            <%= "welcome " + login.name%>
        <% }) %>
        </h1>
    <h2> Please Give us a moment while we are getting ready everthing for you
</h2>

    </tbody>
</table>
<button class="logout" formaction="/warning">Logout</button>
<button class="proceed" id="proceed" formaction="/data">Proceed</button>
</body>
</html>
```

Evaluation sheet:

Roll No	Technology	Max Marks	Marks Awarded	Total (30)
18117	HTML	10		
	CSS	10		
18136	HTML	10		
	CSS	10		
18137	HTML	10		
	CSS	10		
18138	HTML	10		
	CSS	10		
18148	HTML	10		
	CSS	10		
	Project Documentation	10		

B. Tech. project Grace Marks Allocator– G 22

PROJECT REPORT-II

Submitted by

S. No	Name	Roll Number
1.	G Lokesh	CB.EN.U4CSE18117
2.	P Pruthve Rej	CB.EN.U4CSE18136
3.	N Pavan Kruthik	CB.EN.U4CSE18137
4.	P Upendra	CB.EN.U4CSE18138
5.	Sk Awez	CB.EN.U4CSE18148

is fulfilment of the requirements for the Course – 15CSE376 (Net Centric Programming)

**BACHELOR OF TECHNOLOGY IN
COMPUTER SCIENCE AND ENGINEERING**



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING**

AMRITA SCHOOL OF ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE -641112

October 2021

Table of Contents

S.No	Section Name	Page Number
1	JDBC	3
2	Servlet	4
3	ER diagram	6
4	Architecture diagram	7
5	Table specifications	8
6	Functionalities using servlets	11
7	Database connectivity	12
8	Validation	14

Technologies Learnt:

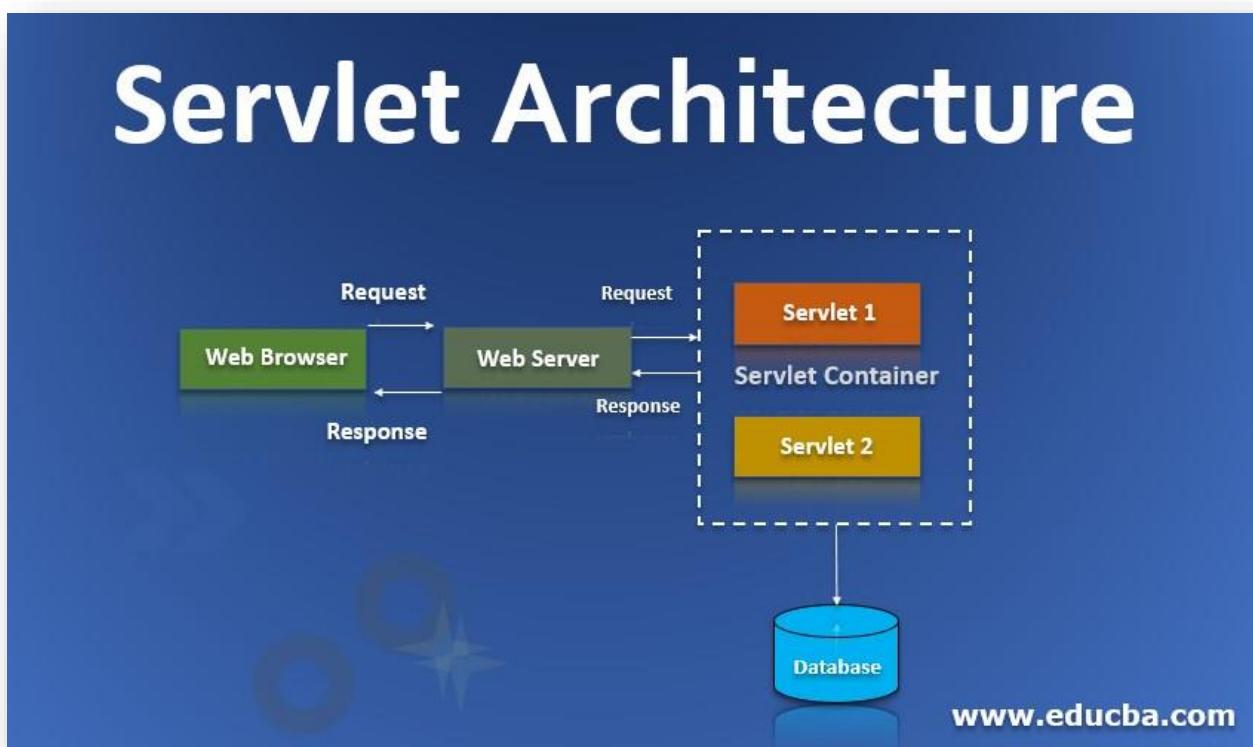
Servlets

Servlets are the Java programs that run on the Java-enabled web server or application server

They are used to handle the request obtained from the web server, process the request, produce the response, then send a response back to the web server.

Properties of Servlets are as follows:

- Servlets work on the server-side.
- Servlets are capable of handling complex requests obtained from the web server.



Execution of Servlets involves six basic steps:

1. The clients send the request to the web server.
2. The web server receives the request.
3. The web server passes the request to the corresponding servlet.
4. The servlet processes the request and generates the response in the form of output.
5. The servlet sends the response back to the web server.
6. The web server sends the response back to the client and the client browser displays it on the screen.

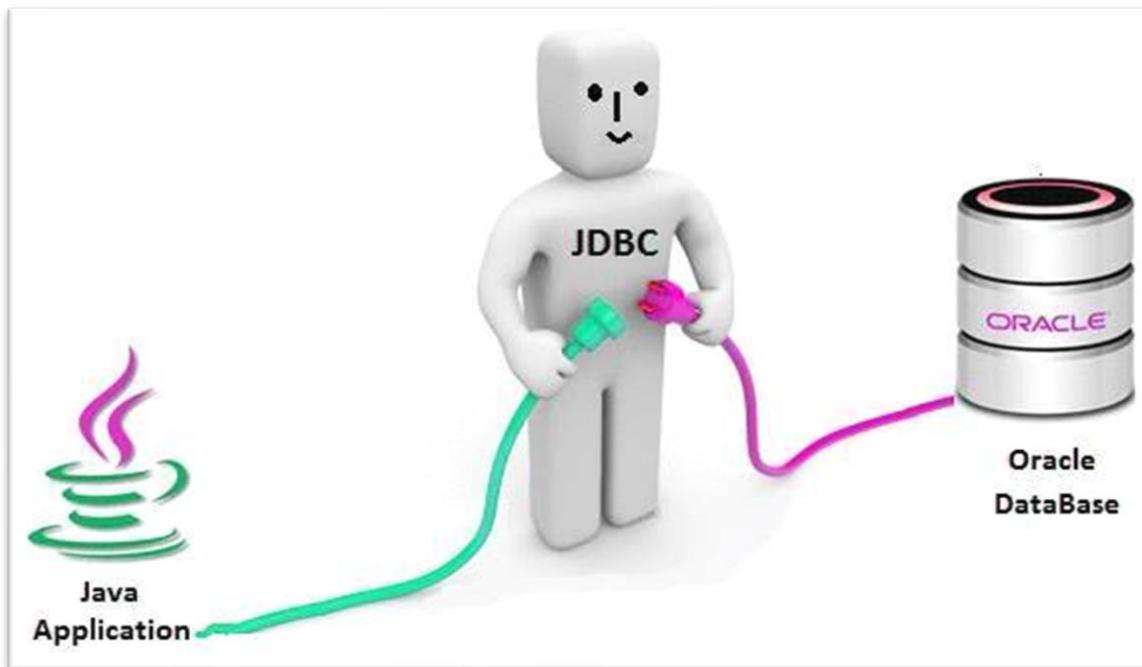
JDBC

JDBC stands for Java Database Connectivity. JDBC is a Java API to connect and execute the query with the database. It is a part of Java SE (Java Standard Edition). JDBC API uses JDBC drivers to connect with the database.

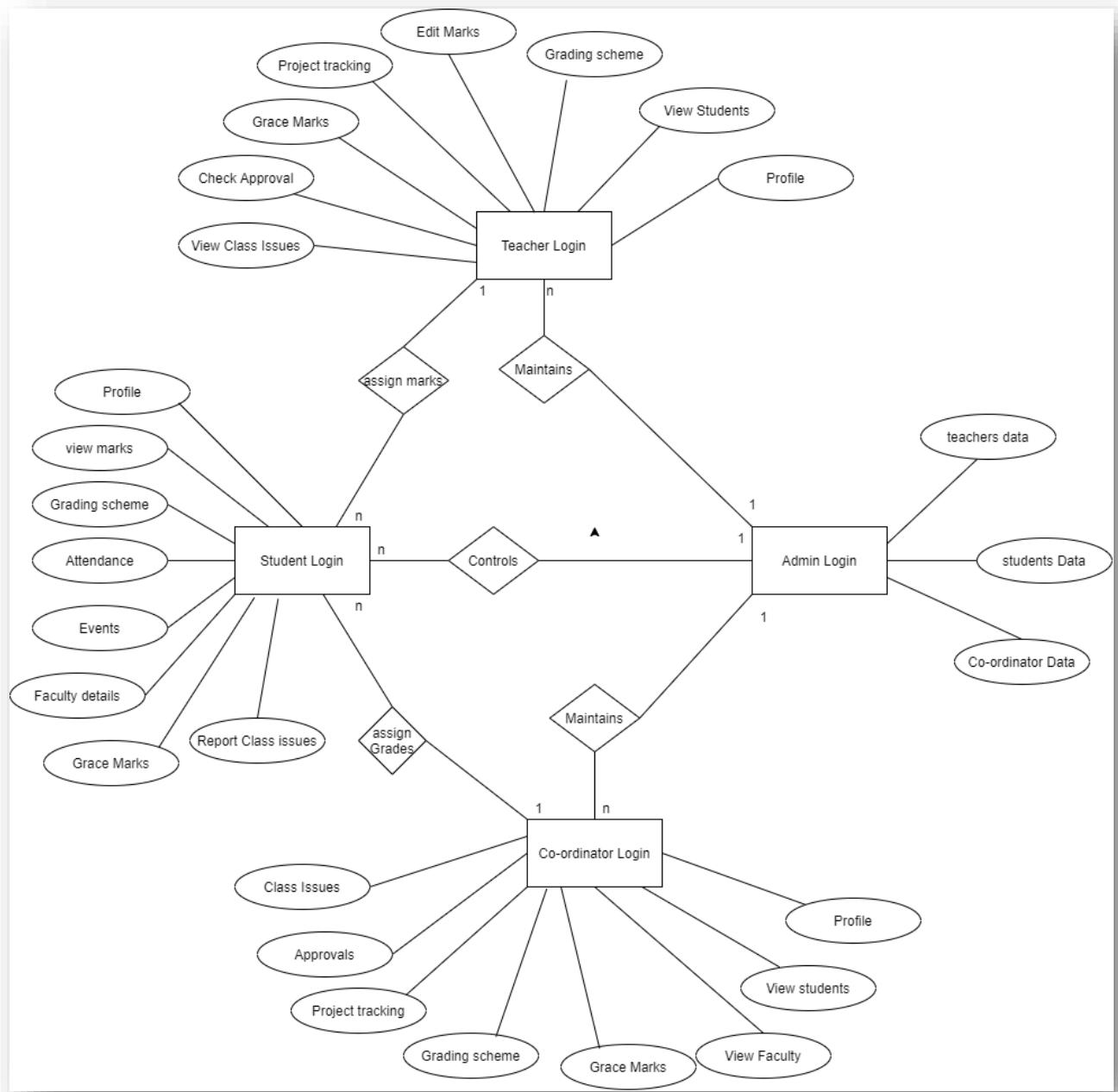
We can use JDBC API to access tabular data stored in any relational database. With the help of JDBC API, we can save, update, delete and fetch data from the database.

We can use JDBC API to handle database using Java program and can perform the following activities:

1. Connect to the database
2. Execute queries and update statements to the database
3. Retrieve the result received from the database.



ER Diagram:



Architecture diagram:

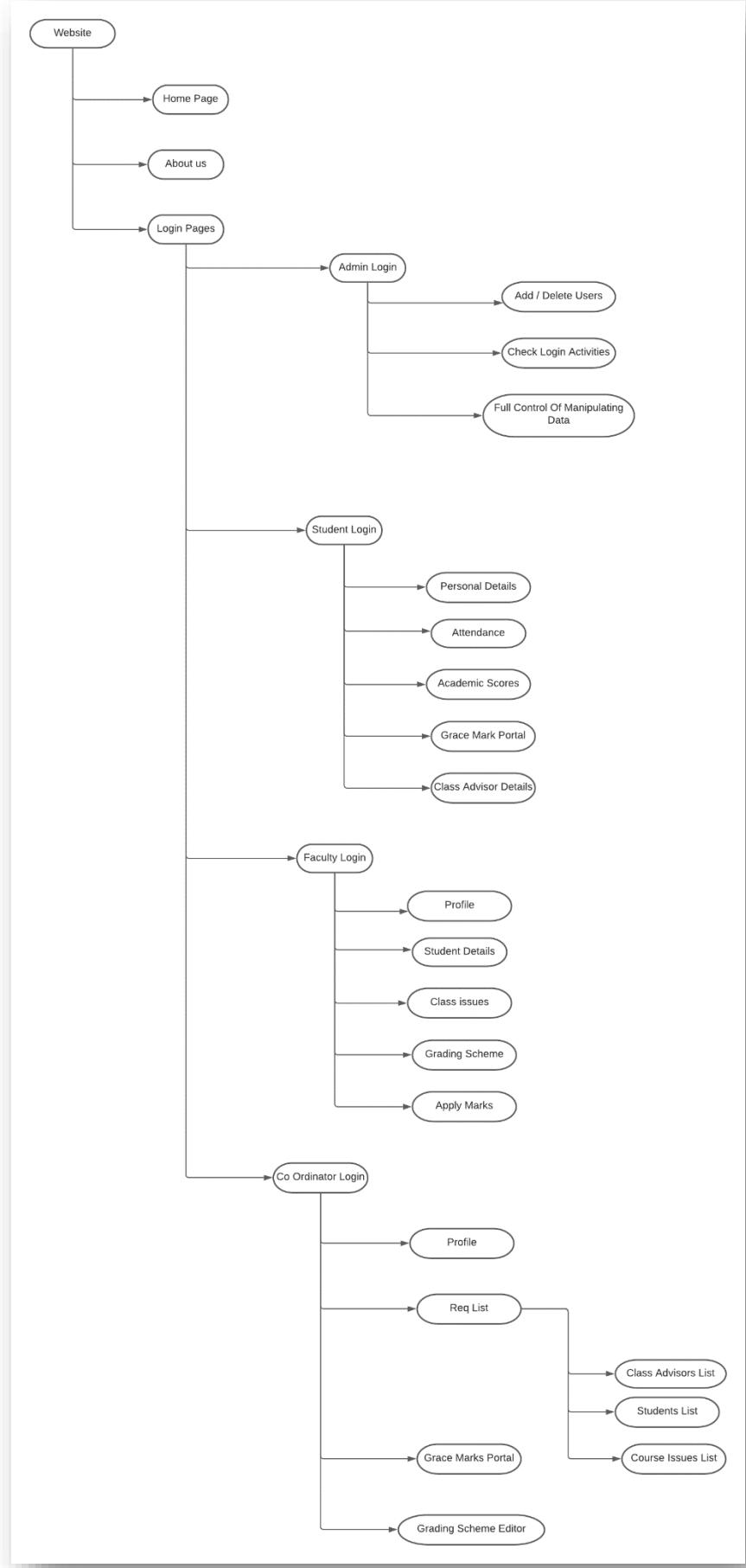


Table Specifications:

1)Users

	Table Name: <input type="text" value="login"/>	Schema: database1								
	Charset/Collation: <input type="text" value="utf8mb4"/> <input type="text" value="utf8mb4_0900_ai_ci"/>	Engine: InnoDB								
	Comments: <input type="text"/>									
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
uname	VARCHAR(20)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
pwd	VARCHAR(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
name	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Column Name:	<input type="text" value="name"/>									
Charset/Collation:	<input type="text" value="Default Charset"/>	<input type="text" value="Default Collation"/>								

2)Profile

	Table Name: <input type="text" value="profile"/>	Schema: database1								
	Charset/Collation: <input type="text" value="utf8mb4"/> <input type="text" value="utf8mb4_0900_ai_ci"/>	Engine: InnoDB								
	Comments: <input type="text"/>									
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
rollno	VARCHAR(30)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
username	TEXT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
dob	TEXT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
mobile	BIGINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
cgpa	DOUBLE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
projects	TEXT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
images	TEXT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
remarks	TEXT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
mother	TEXT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
father	TEXT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
parentmobile	BIGINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
address	TEXT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
mail	TEXT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
year	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
section	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
Column Name:	<input type="text" value="type"/>									
Charset/Collation:	<input type="text" value="Default Charset"/>	<input type="text" value="Default Collation"/>								

3) Approval

4) Project Tracking

5) Otp

6) Attendance

Functionalities using servlets

1) Login Action

- a) Path :/login
- b) Method: Get
- c) Input: username, password

2) Enroll Student

- a)Path:/enroll
- b)Method:Get
- c)Input:Students Data

3) Marks Re enter

- a)Path:/editmarks
- b)Method:Get
- c)Input:Marks Record

4) Approval

- a)Path:/approval
- b)Method:Get
- c)Input:Co Ordinator Approval

5) Class Issues

- a)Path:/issue
- b)Method:Get
- c)Input:Students Feedback

6) Admin Access

- a)Path:/admin
- b)Method:Get
- c)Input:Login Credentials

7) Block User

- a)Path:/block
- b)Method:Get
- c)Input:Reamark , Rollno

8) Event Mark

- a)Path:/graceedit
- b)Method:Get
- c)Input:Mark , Eventype

9) Grade

- a)Path:/Grade
- b)Method:Get
- c)Input:Mark , Subcode

Database connectivity

Java:

```
System.out.println("\n\n***** MySQL JDBC Connection Testing *****");
Connection conn = null;
try
{
    Class.forName ("com.MySQL.jdbc.Driver").newInstance ();
    String userName = "root";
    String password = "Awez@0987";
    String url = "jdbc:MySQL://localhost/database1";
    conn = DriverManager.getConnection (url, userName, password);
    System.out.println ("\nDatabase Connection Established...");
}
catch (Exception ex)
{
    System.err.println ("Cannot connect to database server");
    ex.printStackTrace();
}
```

Javascript:

```
var mysql = require('mysql');
var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "Awez@0987"
});

con.connect(function(err) {
  if (err) throw err;
  console.log("Connected!");
});
```

Insertion:

```
String query="insert into approval (roll , Event , status) values (?,?,?,?,?,?)";
PreparedStatement st=con.prepareStatement(query);
st.setString(1,rollno);
st.setString(2,event);
st.setString(3,status);
```

Updation:

```
String query="UPDATE database1.marks SET "+ subcode +" = ? WHERE (rollno = ?) ";
PreparedStatement st=con.prepareStatement(query);

st.setString(1,mark);
st.setString(2,rollno);
int check=st.executeUpdate();
return check;
```

Validation:

```
String username= req.getParameter("username");
String password= req.getParameter("password");

LoginDao dao = new LoginDao();

try {
    if(dao.connectdb(username, password)) {
        HttpSession session = req.getSession();
        res.sendRedirect("http://localhost:8000/Student");
    }else {
        res.sendRedirect("http://localhost:8000/?username=&password=");
    }
}
```

Validation

Evaluation sheet:

Roll No	Technology	Max Marks	Marks Awarded	Total (30)
18117	Servlet JDBC	10 10		
18136	Servlet JDBC	10 10		
18137	Servlet JDBC	10 10		
18138	Servlet JDBC	10 10		
18148	Servlet JDBC	10 10		
	Project Documentation	10		

B. Tech. project Grace Marks Allocator– G 22

PROJECT REPORT-III

Submitted by

S. No	Name	Roll Number
1.	G Lokesh	CB.EN.U4CSE18117
2.	P Pruthve Rej	CB.EN.U4CSE18136
3.	N Pavan Kruthik	CB.EN.U4CSE18137
4.	P Upendra	CB.EN.U4CSE18138
5.	Sk Awez	CB.EN.U4CSE18148

is fulfilment of the requirements for the Course – 15CSE376 (Net Centric Programming)

BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING**

AMRITA SCHOOL OF ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE -641112

November 2021

Table of Contents

S.No	Section Name	Page Number
1	Abstract	3
2	Technologies Used	4
3	Implementation	6
4	Ouput	7
5	Validation	8
6	Evaluation	11

Abstract

Technologies Used

HTML

HTML is the language for describing the structure of Web pages. HTML gives authors the means to:

- Publish online documents with headings, text, tables, lists, photos, etc.
- Retrieve online information via hypertext links, at the click of a button.
- Design forms for conducting transactions with remote services, for use in searching for information, making reservations, ordering products, etc.
- Include spread-sheets, video clips, sound clips, and other applications directly in their documents

With HTML, authors describe the structure of pages using markup. The elements of the language label pieces of content such as “paragraph,” “list,” “table,” and so on

CSS:

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML. CSS describes how elements should be rendered on screen, on paper, in speech, or on other media. CSS is used to style and layout web pages — for example, to alter the font, colour, size, and spacing of your content, split it into multiple columns, or add animations and other decorative features. This module provides a gentle beginning to your path towards CSS mastery with the basics of how it works, what the syntax looks like, and how you can start using it to add styling to HTML.

JavaScript:

JavaScript (JS) is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat. JavaScript is a prototype-based, multi-paradigm, single-threaded, dynamic language, supporting object-oriented, imperative, and declarative styles.

Embedded JavaScript:

You can embed JavaScript in an HTML document in the following ways: As statements and functions within a <SCRIPT> tag. ... See "Specifying a file of JavaScript code". By specifying a JavaScript expression as the value for an HTML attribute. See "Using JavaScript expressions as HTML attribute values".

Servlets

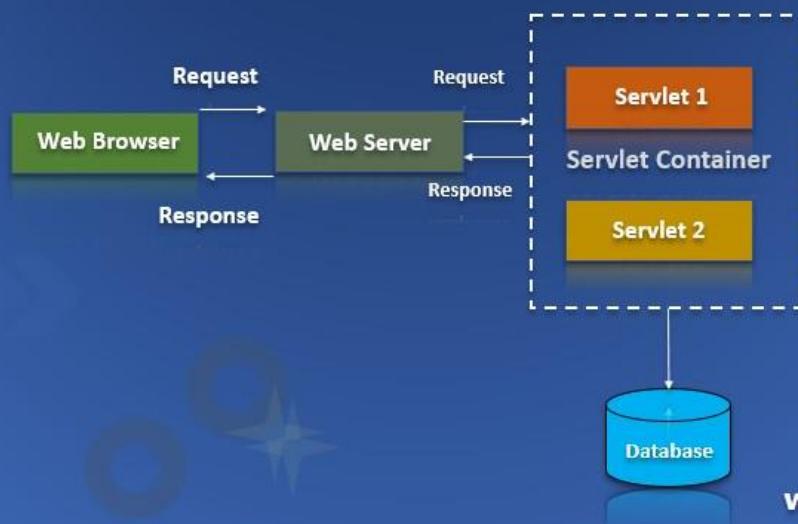
Servlets are the Java programs that run on the Java-enabled web server or application server

They are used to handle the request obtained from the web server, process the request, produce the response, then send a response back to the web server.

Properties of Servlets are as follows:

- Servlets work on the server-side.
- Servlets are capable of handling complex requests obtained from the web server.

Servlet Architecture



www.educba.com

Execution of Servlets involves six basic steps:

1. The clients send the request to the web server.
2. The web server receives the request.
3. The web server passes the request to the corresponding servlet.
4. The servlet processes the request and generates the response in the form of output.
5. The servlet sends the response back to the web server.
6. The web server sends the response back to the client and the client browser displays it on the screen.

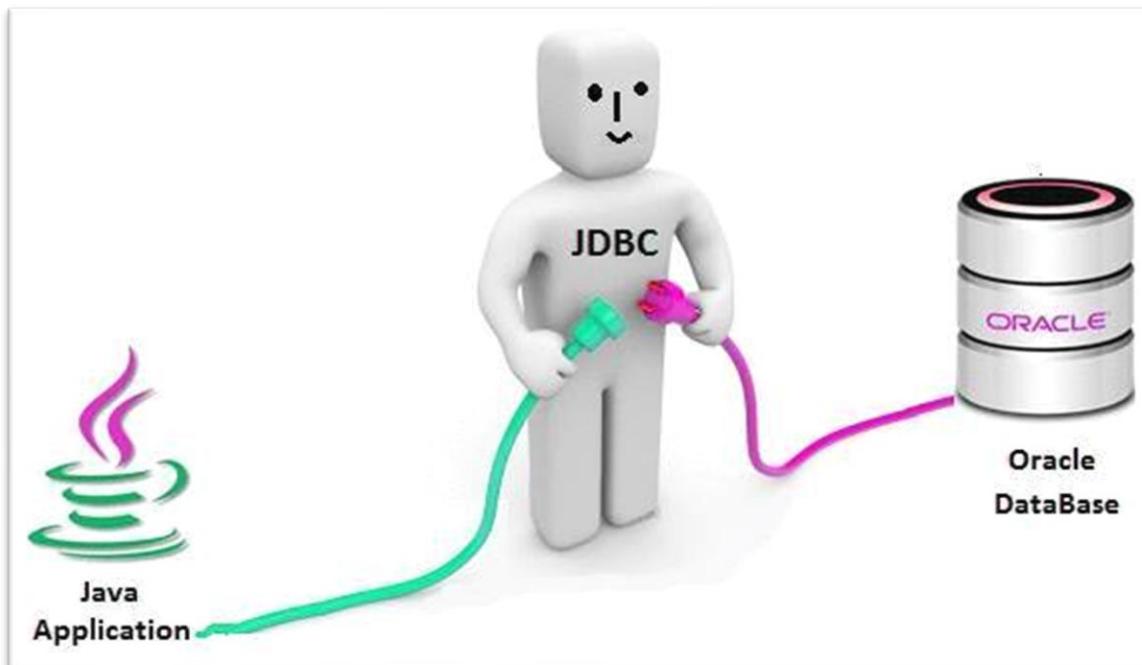
JDBC

JDBC stands for Java Database Connectivity. JDBC is a Java API to connect and execute the query with the database. It is a part of Java SE (Java Standard Edition). JDBC API uses JDBC drivers to connect with the database.

We can use JDBC API to access tabular data stored in any relational database. With the help of JDBC API, we can save, update, delete and fetch data from the database.

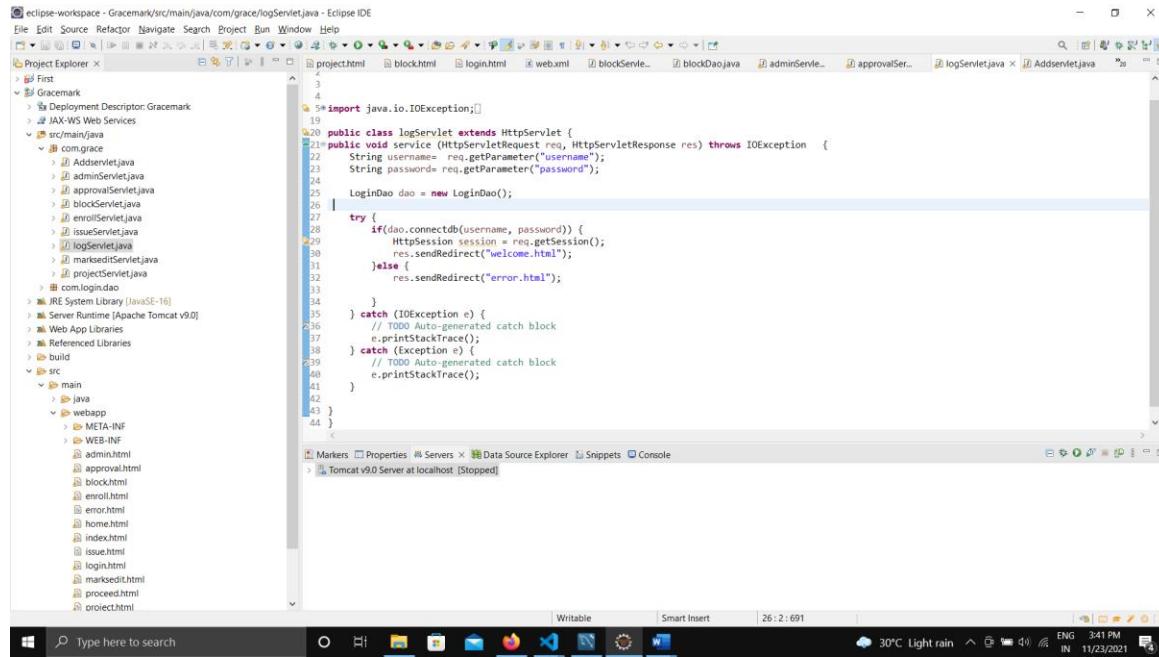
We can use JDBC API to handle database using Java program and can perform the following activities:

1. Connect to the database
2. Execute queries and update statements to the database
3. Retrieve the result received from the database.



Implementation

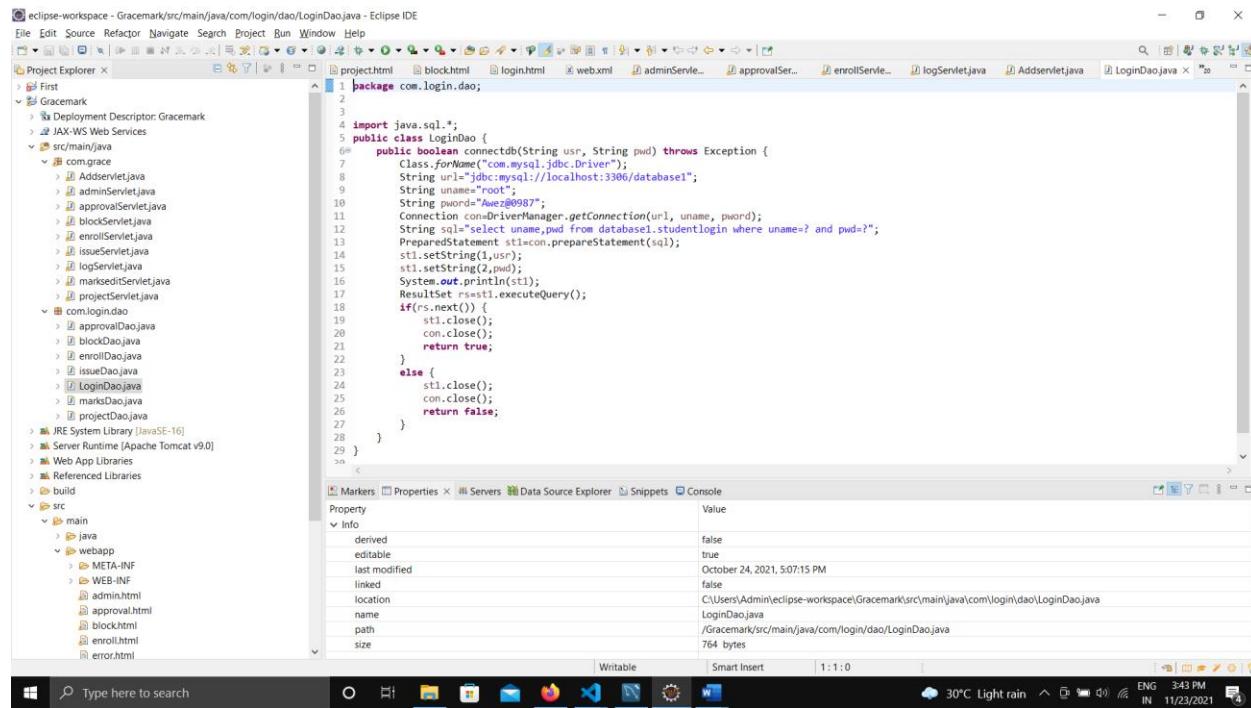
Login Servlet :



```
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer X
First
Gracemark
Deployment Descriptor: Gracemark
JAX-WS Web Services
src/main/java
com.grace
Addserver.java
adminServlet.java
approvalServlet.java
blockServlet.java
enrollServlet.java
issueServlet.java
logServlet.java
markedServlet.java
projectServlet.java
com.login.java
I RE System Library [JavaSE-16]
Server Runtime [Apache Tomcat v9.0]
Web App Libraries
Referenced Libraries
build
src
main
java
webapp
META-INF
WEB-INF
admin.html
approval.html
block.html
enroll.html
error.html
home.html
index.html
issue.html
login.html
marked.html
proced.html
project.html
Markers Properties Servers Data Source Explorer Snippets Console
Tomcat v9.0 Server at localhost [Stopped]
Writable Smart Insert 26 : 2 : 691
30°C Light rain ENG 3:41 PM IN 11/23/2021
```

```
import java.io.IOException;
public class logServlet extends HttpServlet {
    public void service (HttpServletRequest req, HttpServletResponse res) throws IOException {
        String username= req.getParameter("username");
        String password= req.getParameter("password");
        LoginDao dao = new LoginDao();
        try {
            if(dao.connectdb(username, password)) {
                HttpSession session = req.getSession();
                res.sendRedirect("welcome.html");
            }else {
                res.sendRedirect("error.html");
            }
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        } catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```

JDBC:



```
File Edit Source Refactor Navigate Search Project Run Window Help
Project Explorer X
First
Gracemark
Deployment Descriptor: Gracemark
JAX-WS Web Services
src/main/java
com.grace
Addserver.java
adminServlet.java
approvalServlet.java
blockServlet.java
enrollServlet.java
issueServlet.java
logServlet.java
markedServlet.java
projectServlet.java
com.login.dao
approvalDao.java
blockDao.java
enrollDao.java
issueDao.java
LoginDao.java
markDao.java
projectDao.java
I RE System Library [JavaSE-16]
Server Runtime [Apache Tomcat v9.0]
Web App Libraries
Referenced Libraries
build
src
main
java
webapp
META-INF
WEB-INF
admin.html
approval.html
block.html
enroll.html
error.html
login.html
marked.html
project.html
Markers Properties Servers Data Source Explorer Snippets Console
Property Value
Info
derived false
editable true
last modified October 24, 2021, 5:07:15 PM
linked false
location C:/Users/Admin/eclipse-workspace/Gracemark/src/main/java/com/login/dao/LoginDao.java
name LoginDao.java
path /Gracemark/src/main/java/com/login/dao/LoginDao.java
size 764 bytes
Writable Smart Insert 1:1:0
30°C Light rain ENG 3:43 PM IN 11/23/2021
```

```
package com.login.dao;
import java.sql.*;
public class LoginDao {
    public boolean connectdb(String usr, String pwd) throws Exception {
        Class.forName("com.mysql.jdbc.Driver");
        String url="jdbc:mysql://localhost:3306/database1";
        String uname="root";
        String pword="Avez@0987";
        Connection con=DriverManager.getConnection(url, uname, pword);
        String sql="select uname,pwd from database1.studentlogin where uname=? and pwd=?";
        PreparedStatement st1=con.prepareStatement(sql);
        st1.setString(1,usr);
        st1.setString(2,pwd);
        System.out.println(st1);
        ResultSet rsst1.executeQuery();
        if(rsst1.next()) {
            st1.close();
            con.close();
            return true;
        } else {
            st1.close();
            con.close();
            return false;
        }
    }
}
```

Enroll Servlet :

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer**: Shows the project structure with files like First, Gracemark, Deployment Descriptor, JAX-WS Web Services, src/main/java, com.grace, and com.login.dao.
- Code Editor**: Displays the Java code for `enrollServlet.java`. The code handles user input from a form and calls methods from `enrollDao` to save data to a database.
- Properties View**: Shows the properties for the selected file (`enrollServlet.java`). Key properties include:
 - derived: false
 - editable: true
 - last modified: October 30, 2021, 20:15 PM
 - linked: false
 - location: C:\Users\Admin\ecipse-workspace\Gracemark\src\main\java\com\grace\enrollServlet.java
 - name: enrollServlet.java
 - path: /Gracemark/src/main/java/com/grace/enrollServlet.java
 - size: 1,904 bytes
- Bottom Bar**: Includes the Windows taskbar with icons for File Explorer, Mail, Task View, and Start.

JDBC:

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer View:** Shows the project structure. The main package is com.login.dao, containing EnrollDao.java and several other DAO classes like ApprovalDao.java, BlockDao.java, etc.
- Editor View:** Displays the Java code for EnrollDao.java. The code uses JDBC to insert data into a database table named profile. It includes imports for java.sql.* and javax.sql.DataSource, and uses PreparedStatement to execute the SQL query.
- Properties View:** Shows the properties for the com.login.dao project. Key details include:
 - Info:** derived (false), editable (true), last modified (October 24, 2021, 8:03:22 PM).
 - Location:** C:/Users/Admin/eclipse-workspace/Gracemark/src/main/java/com/login/dao/
 - Path:** /Gracemark/src/main/java/com/login/dao/EnrollDao.java
 - Size:** 1,400 bytes
- Bottom Status Bar:** Shows the date (11/23/2021), time (3:44 PM), and weather (30°C Light rain).

Project Servlet :

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure under "First".
- Code Editor:** Displays the Java code for `projectServlet.java`. The code handles HTTP requests and interacts with a database via DAO objects.
- Properties View:** Shows properties for `projectServlet.java`, including derived (false), editable (true), last modified (October 30, 2021, 2:02:15 PM), linked (false), location (C:/Users/Admin/eclipse-workspace/Gracemark/src/main/java/com/grace/projectServlet.java), name (projectServlet.java), path (/Gracemark/src/main/java/com/grace/projectServlet.java), and size (1,483 bytes).
- OS Taskbar:** Shows system icons, weather (30°C Light rain), and system status (ENG IN 11/23/2021).

JDBC:

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure under "First".
- Code Editor:** Displays the Java code for `projectDao.java`. The code contains JDBC code for inserting data into a MySQL database.
- Properties View:** Shows properties for `projectDao.java`, including derived (false), editable (true), last modified (October 30, 2021, 12:38:09 PM), linked (false), location (C:/Users/Admin/eclipse-workspace/Gracemark/src/main/java/com/login/dao/projectDao.java), name (projectDao.java), path (/Gracemark/src/main/java/com/login/dao/projectDao.java), and size (1,060 bytes).
- OS Taskbar:** Shows system icons, weather (30°C Light rain), and system status (ENG IN 11/23/2021).

Approval

The screenshot shows the Eclipse IDE interface with the approvalServlet.java file open in the editor. The code implements a servlet that handles a POST request to approve a user. It connects to a database, updates the status, and sends a success or error redirect. The code includes several TODO comments for logging.

```
1 package com.grace;
2 import java.io.IOException;
3
4 public class approvalServlet extends HttpServlet {
5     public void service (HttpServletRequest req, HttpServletResponse res) throws IOException {
6         String rollno= req.getParameter("rollno");
7         String event= req.getParameter("event");
8         String status= req.getParameter("status");
9         approvalDao obj = new approvalDao();
10
11         try {
12             int count= obj.connectdb(rollno, event, status);
13             if(count==1) {
14                 res.sendRedirect("success.html");
15             }
16         } catch (IOException e) {
17             // TODO Auto-generated catch block
18             e.printStackTrace();
19         } catch (Exception e) {
20             // TODO Auto-generated catch block
21             e.printStackTrace();
22         }
23     }
24 }
```

The Properties view shows the following details for the approvalServlet.java file:

Property	Value
derived	false
editable	true
last modified	October 30, 2021, 2:01:40 PM
linked	false
location	C:/Users/Admin/eclipse-workspace/Gracemark/src/main/java/com/grace/approvalServlet.java
name	approvalServlet.java
path	/Gracemark/src/main/java/com/grace/approvalServlet.java
size	1,127 bytes

Block User

The screenshot shows the Eclipse IDE interface with the blockServlet.java file open in the editor. The code implements a servlet that handles a POST request to block a user. It connects to a database, updates the status, and sends a success or error redirect. The code includes several TODO comments for logging.

```
1 package com.grace;
2
3 public class blockServlet extends HttpServlet {
4     public void service (HttpServletRequest req, HttpServletResponse res) throws IOException {
5         String uname= req.getParameter("rollno");
6         String remark= req.getParameter("remark");
7         String logintype= req.getParameter("logintype");
8         String status= req.getParameter("status");
9         blockDao obj = new blockDao();
10
11         try {
12             int count= obj.blockuser(uname, remark, logintype , status);
13             if(count==1) {
14                 res.sendRedirect("success.html");
15             }
16         } catch (IOException e) {
17             // TODO Auto-generated catch block
18             e.printStackTrace();
19         } catch (Exception e) {
20             // TODO Auto-generated catch block
21             e.printStackTrace();
22         }
23     }
24 }
```

The Properties view shows the following details for the blockServlet.java file:

Property	Value
derived	false
editable	true
last modified	October 30, 2021, 2:01:50 PM
linked	false
location	C:/Users/Admin/eclipse-workspace/Gracemark/src/main/java/com/grace/blockServlet.java
name	blockServlet.java
path	/Gracemark/src/main/java/com/grace/blockServlet.java
size	1,173 bytes

Active Login :

```
<div class="container">
<table class="table table-striped">
<% activelogins.forEach(function (login) { %>
<tr>
  <td><%= login.username %>      </td>
  <td><%= login.rollno %>      </td>
  <td> <%= login.mobile %>      </td>
  <td> <%= login.mail %>      </td>
  <td> </td>
</tr>

<% }) %>
```

Attendance:

```
<% attendance.forEach(function (students) { %>
<tr>
  <th> Course Code </th>
  <th> Course Name </th>
  <th> Attendace in % </th>
</tr>
<tr>
  <td>15CSE376</td>
  <td>Net Centric Programming</td>
  <td ><%=students.NCP%> </td></tr>

  <td>15CSE401</td>
  <td>Machine Learning and Data Mining</td>
  <td ><%=students.MLDM%> </td></tr>

  <td>15CSE402</td>
  <td>Structure and Interpretation of Computer Programs</td>
  <td ><%=students.SICP%> </td></tr>
  <td>15CSE480</td>
  <td>Internet of Things</td>
  <td ><%=students.IOT%> </td></tr>
  <td>15CSE481</td>
  <td>Machine Learning and Data Mining Lab.</td>
  <td ><%=students.MLDMLAB%> </td></tr>
```

```
<td>15CSE495</td>

<td>Project Phase I</td>
<td ><%=students.PP1%> </td></tr> <% } ) %>
</center>
```

Retrieving Student Details

```
var obj1 = {};
app.set('views',path.join(__dirname,'views'));
app.set('view engine', 'ejs');
app.get('/students', function(req, res) {
fs.readFile('currentlogin.txt', 'utf-8', (err, data) => {
if (err) throw err;
mycon.query(` SELECT * FROM database1.profile where type="student"; `,
function(err, result) {
    if(err){
        throw err;
    } else {
        obj1 = {students_details: result};
        res.render('students_details', obj1);
// console.log(obj1);
    }
});
});
});
```

Project Dashboard

```
app.get('/trackingdashboard', function(req, res) {
fs.readFile('currentlogin.txt', 'utf-8', (err, data) => {
if (err) throw err;
mycon.query(` SELECT * FROM database1.tracking , profile where (profile.username = tracking.guide)` , function(err, result) {
if(err){
throw err;
} else {
obj1 = {trackingdashboard: result};
res.render('trackingdashboard', obj1);
// console.log(obj1);
}
});
});
});
```

Faculty Details

```
app.get('/faculty_details', function(req, res) {
var obj2=[];
mycon.query(` SELECT * FROM database1.profile where type="faculty"; ` ,
function(err, result) {
if(err){
throw err;
} else {
obj2 = {faculty_details: result};
res.render('faculty_details', obj2);
}});
});
```

Profile

```
fs.readFile('logintype.txt', 'utf-8', (err, data1) => {
  fs.readFile('currentlogin.txt', 'utf-8', (err, data) => {
    if (err) throw err;
    mycon.query(`SELECT * FROM profile where (profile.rollno = '${data}')`, function(err, result) {

      if(err){
        throw err;
      } else {
        obj = {upload: result};

        res.render('upload', obj);
      }
    });
  });
})
})
```

Erasing Current user from database when logged out

```
app.get('/logout', function(req, res){
  res.sendfile("logout.html");
  fs.readFile('currentlogin.txt', 'utf-8', (err, data) => {
    if (err) throw err;
    mycon.query(`truncate table database1.currentlogin`,function(err,result){
      if(err) throw err;
      // console.log(result);
    });
  });
})
```

Retrieving Marks

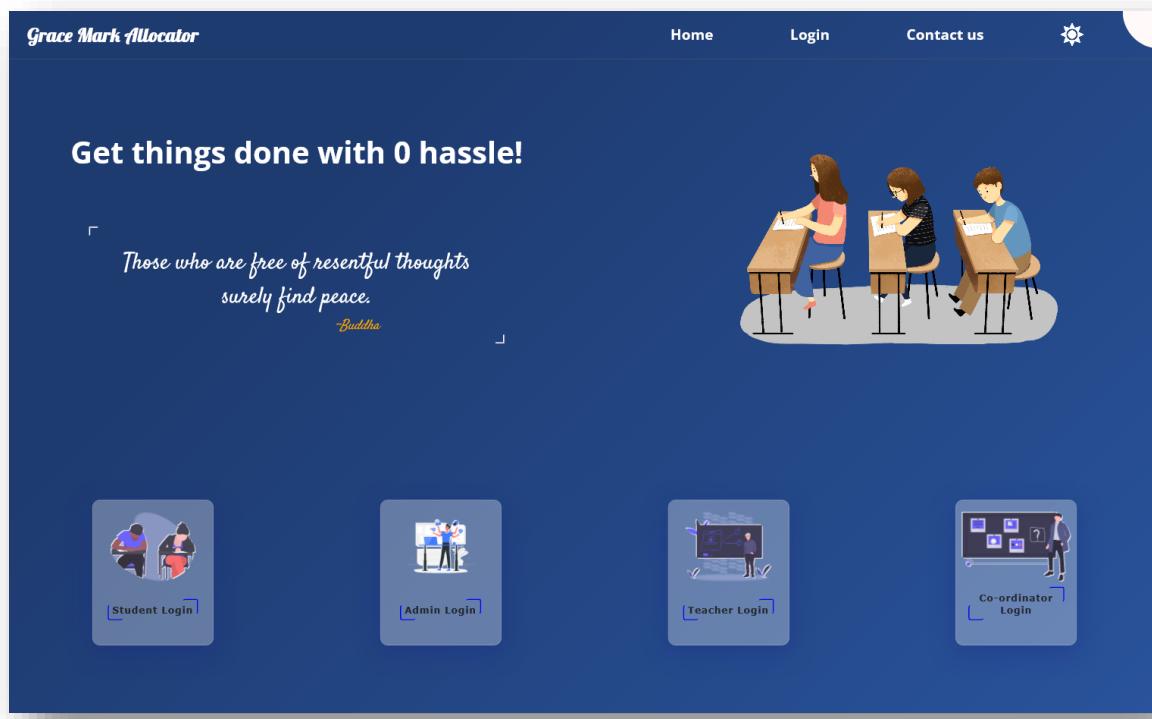
```
app.get('/selfmarks', function(req, res) {
var obj2={}
fs.readFile('currentlogin.txt', 'utf-8', (err, data) => {
if (err) throw err;
mycon.query(` SELECT * FROM database1.marks where rollno =  '${data}'`, 
function(err, result) {
if(err){
throw err;
} else {
obj2 = {marks: result};
res.render('marks', obj2);
}});
});
```

Uploading Documents Function

```
http.createServer(function (req, res) {
  if (req.url == '/fileupload') {
    var form = new formidable.IncomingForm();
    form.parse(req, function (err, fields, files) {
      var oldpath = files.filetoupload.path;
      var newpath = 'C:/Users/Admin/Desktop/Ncp/Project/Upload/' +
files.filetoupload.name;
      fs.rename(oldpath, newpath, function (err) {
        if (err) throw err;
        res.write('File uploaded and moved!');
        res.end();
      });
    });
  } else {
    res.writeHead(200, {'Content-Type': 'text/html'});
    res.write('<form action="fileupload" method="post" enctype="multipart/form-
data">');
    res.write('<input type="file" name="filetoupload"><br>');
    res.write('<input type="submit">');
    res.write('</form>');
    return res.end();
  }
}).listen(8080);
```

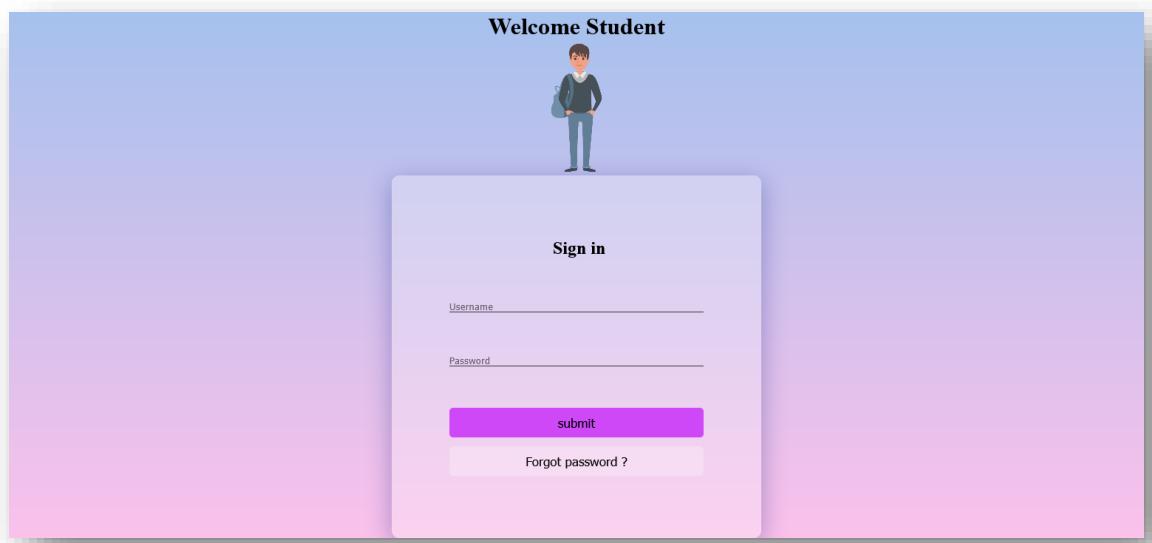
Output

Home page:



The screenshot shows the homepage of the "Grace Mark Allocator" application. The header features the title "Grace Mark Allocator" on the left and navigation links for "Home", "Login", and "Contact us" on the right. A sun icon is also present in the top right corner. The main content area has a dark blue background. At the top, a quote by Buddha is displayed: "Those who are free of resentful thoughts surely find peace." Below the quote, there is a cartoon illustration of three students sitting at desks and writing. The central part of the page contains four login buttons: "Student Login" (with two students icon), "Admin Login" (with two people at a computer icon), "Teacher Login" (with a teacher at a chalkboard icon), and "Co-ordinator Login" (with a person at a large screen icon). The overall design is clean and modern.

Login :



The screenshot shows the "Welcome Student" login page. The background is a gradient from light blue at the top to pink at the bottom. In the center, there is a white rectangular form. At the top of the form, it says "Welcome Student" and features a small cartoon character of a student standing. Below this, the word "Sign in" is centered. There are two input fields: one for "Username" and one for "Password", both with placeholder text. Below the password field is a purple "submit" button. At the bottom of the form, there is a link "Forgot password ?". The overall design is user-friendly and visually appealing.

Welcome page:

Grace mark for publication

Hello, shaik awez

Logout

☰ Menu



Menu:

Profile

View Marks

Grading Scheme

Attendance

Events

Report Class Issues

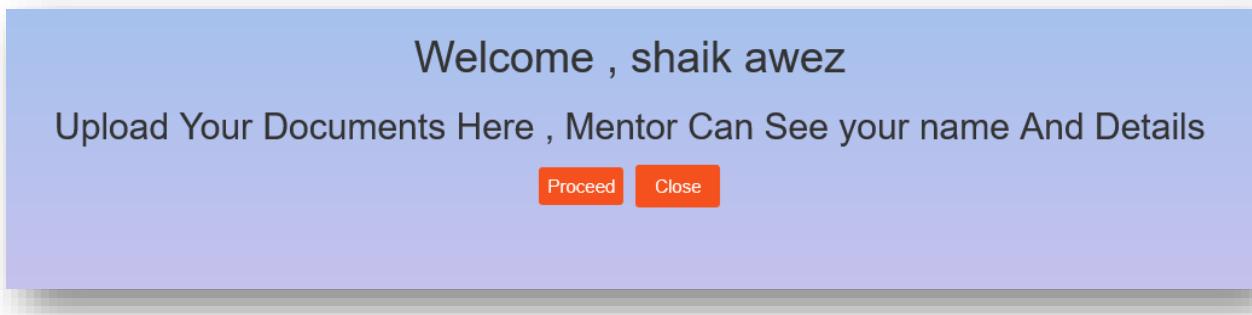
Grace Marks

Project Upload

Logout



Doc Uploading:



Active logins:

Active logins					Home	Block	History	Refresh	Logout
Username	Rollno	Mobile	Mail	Images					
upendra	cb.en.u4cse18138	9182410276	ramt09444@gmail.com						

User Block

Block / Unblock User

[Logout](#)



Roll no :

Remark :

User : Faculty Student Co ordinator

Action : Block Unblock

Login History:



Students Data:

Students Data											Refresh	Close
Roll no	Name	DOB	Mobile	CGPA	Mother	Father	Parent Mobile	Address	G-Mail ID	Photo		
cb.en.u4cse18117	Lokesh	12/9/2000	8142838171	7.5	sudha rani	srinivasulu	9246477785	Narasaraopet,Guntur,AP	garlapatilokesh16@gmail.com			
cb.en.u4cse18136	pruthve	31-08-2001	9381044045	7	Raji	Ravi	9032702362	trvineni talkies ,Ananthpur ,AP	pruthverej1947@gmail.com			
cb.en.u4cse18137	pavan	24-10-2000	834190089	7.2	Geetha	Vinod	9440924291	Rajeev nagar extension,Tirupati,Andhra Pradesh	Kruthikpavan24@gmail.com			
cb.en.u4cse18138	upendra	20-10-2000	9182410276	7.3	devi	Sivaiah	9704108768	nrt., Guntur district Andhra Pradesh	ramt09444@gmail.com			
cb.en.u4cse18148	shaik awez	23-12-2000	6303731463	7	shaik Habeeb	shaik Jaleel	9441234780	Buchi reddy palem , Nellore,Ap	shaikjallil123@gmail.com			

Project tracking :

Tracking of Paper Publication												
Project id	Title	Team		Dataset	Section	Guide	Approved	Status	Supporting Documents	Contact	More Details	
D231	Covid 19 Prediction Based On Comorbidities	1. Shaik Awez CB.EN.U4CSE18148 2. N pavan Kruthik CB.EN.U4CSE18137 3 P Pruthve Rej CB.EN.U4CSE18136 4 P Upendra CB.EN.U4CSE18138	Kaggale	CSE B	DR.Arun Kumar	Yes	Pre proceesing the data set	Click here	6303731463	Click Here		

More details on Project :

Abstract

A health crisis of massive proportion such as the current COVID-9 pandemic provides us with an opportunity to ponder and reflect over what we can better in the way we deal with healthcare to make us humans be more prepared and enabled to combat such an event in the future. During the entire course of the pandemic, one of the main problems that healthcare providers have faced is the shortage of medical resources and a proper plan to efficiently distribute them. They have been in the dark failing to understand how much resource they could even in the very next week as the COVID-19 curve has swayed very unpredictably. In these tough times, being able to predict what kind of resource an individual might require at the time of being tested positive or even before that will be of great help to the authorities as they would be able to procure and arrange for the resources necessary to save the life of that patient.

Problem Definition

To design a web application for predicting the Covid 19 virus using sampling techniques and Feature selection Methods . The Objective of the project is to design a web based application and to apply AlgoThe Objective of the project is to design a web based application and to apply AlgoThe Objective of the project is to design a web based application and to apply Algorithms to include an optimal sampling technique to accurately predict the virus

Introduction

A health crisis of massive proportion such as the current COVID-9 pandemic provides us with an opportunity to ponder and reflect over what we can better in the way we deal with healthcare to make us humans be more prepared and enabled to combat such an event in the future. During the entire course of the pandemic, one of the main problems that healthcare providers have faced is the shortage of medical resources and a proper plan to efficiently distribute them. They have been in the dark failing to understand how much resource they could even in the very next week as the COVID-19 curve has swayed very unpredictably. In these tough times, being able to predict what kind of resource an individual might require at the time of being tested positive or even before that will be of great help to the authorities as they would be able to procure and arrange for the resources necessary to save the life of that patient

Guide

Roll No	COOD21
Name	DR Arun Kumar
Allotted Sections	All
Branch	CSE,ECE
Graduation Year	2004
Mobile	1234568901
Projects	Prediction of cancer using customised fuzzy rough machine learning approaches ,Rough Set Based Variable Tolerance Attribute Selection on High Dimensional Microarray Imbalanced Data
Mail	c_arunkumar@gmail.com
Address	Coimbatore , Tamil Nadu



Website



Code

Faculty Welcome page:

Hello, DR.Ramesh

Grace mark for publication

Logout

☰ Menu



Menu :

Profile

Grading Scheme

View Students

View Marks

Class Issues

Project tracking

Grace Marks

Approvals

Logout

Grace mark for publication



host:8000/gracemarks
host:8000/gracemarks

Validation

Evaluation sheet:

Roll No	Technology	Max Marks	Marks Awarded	Total (30)
18117	Business Logic Integration	10 10		
18136	Business Logic Integration	10 10		
18137	Business Logic Integration	10 10		
18138	Business Logic Integration	10 10		
18148	Business Logic Integration	10 10		
	Project Documentation	10		