Extracurricular Activities

PROJECT REPORT

Submitted by

REG NO	NAME
CB.EN.U4CSE17420	Gayathri E
CB.EN.U4CSE17424	Prathyusha I
CB.EN.U4CSE17445	Neeraj P Reddy
CB.EN.U4CSE17453	Sanjay Tharagesh R S
CB.EN.U4CSE17458	Srishilesh P S

in partial 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 2020

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF ENGINEERING, COIMBATORE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that the report entitled "**Extracurricular activities**" submitted by

REG NO	**NAME**
CB.EN.U4CSE17420	Gayathri E
CB.EN.U4CSE17424	Prathyusha I
CB.EN.U4CSE17445	Neeraj P Reddy
CB.EN.U4CSE17453	Sanjay Tharagesh R S
CB.EN.U4CSE17458	Srishilesh P S

in partial fulfilment of the requirements for the award of grade in course - 15CSE376 NET CENTRIC PROGRAMMING is a bonafide record of the work carried out at Amrita School of Engineering, Coimbatore.

Evaluated on:

Course Faculty Faculty Examiner

ACKNOWLEDGEMENT

I wish to record my deep sense of gratitude and profound thanks to my course faculty Dr. Harini. N., Asst. Professor, Computer Science and Engineering Department, Amrita School of Engineering, Coimbatore, for her keen interest, inspiring guidance, constant encouragement with my work during all stages, to bring this report into fruition.

I also thank Sabarish B. A., Ramya G. R. (Asst CSE Faculty in NCP lab) for being a support throughout our project period providing us with all the feedback. I express my gratitude to all the lab faculty members mentioned below, Department of ICTS, Coimbatore for being a support throughout our project period providing us with all the resources possible.

Gayathri E

Prathyusha I

Neeraj Kumar P

Sanjay Tharagesh R S

Srishilesh P S

IT BLOCK GF CPLAB - 1	Mr Kuppuraj
IT BLOCK GF CPLAB - 2	Mr Ashwin
IT BLOCK FF CPLAB - 3	Mr Anoop
IT BLOCK FF CPLAB - 4	Mr Sumesh
IT BLOCK SF CPLAB V (VLSI)	Ms Parvathy
IT BLOCK SF CPLAB V (PE)	Mr Jayaram
PG LAB (AB3)	Mr Pradeep

TABLE OF CONTENTS

ABSTRACT

In education institutions, managing various events manually can be hectic work for both organizers and students. There can be a lot of mismanagement and disorder in the information stored and retrieved. Moreover, it is a time-consuming process for organizers to monitor and integrate all the information to reduce data redundancy and generate reports from it. In some cases, it is also difficult for students to reach out to the organizers to check their registration status, upcoming events, and announcements. Thus, we developed an integrated platform called Fiesta, for organizers to help organize events very easily, keep track of student registrations, manage upcoming events, generate reports, search students based on skills and other numerous functionalities that support organizers to complete the essential tasks right from the web portal. This portal will act as a one-stop source for all of the event needs.

Modules

- Landing Page - with student registration and admin login
- Contact and About Us
- Announcement section
- Upcoming Events section
- Event details page
- Event winners announcement
- Student leaderboard
- Search students based on skills
- Page to view student registrations for an event
- Generate Comprehensive Reports based on duration and filters

INTRODUCTION

A web based application which satisfies all the functionalities like student registrations for an event, winner announcement and lot more. Considering the functionalities that this application must contain, the whole application is divided into various modules which includes:

- Landing Page - with student registration and admin login
- Contact and About Us
- Announcement section
- Upcoming Events section
- Event details page
- Event winners announcement
- Student leaderboard
- Search students based on skills
- Page to view student registrations for an event
- Generate Comprehensive Reports based on duration and filters

TERM 1

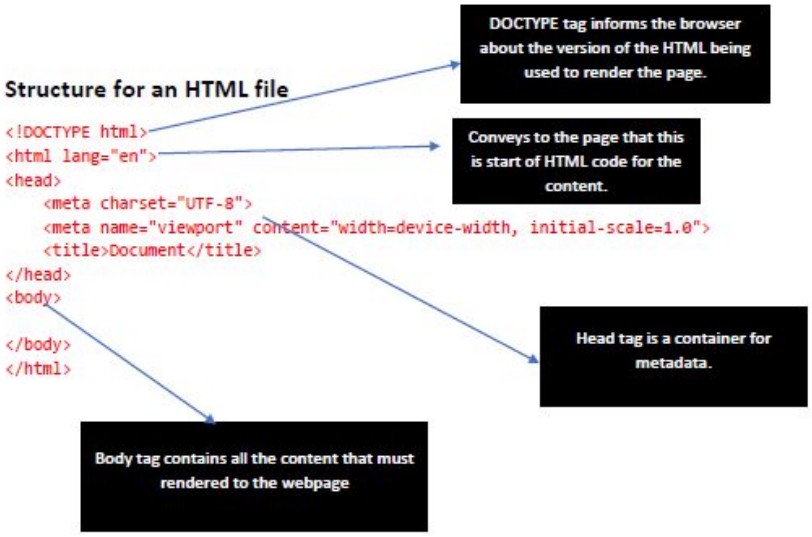
Introduction to Web Designing - HTML5

Hypertext Markup Language(HTML) is standard markup language for documents designed to provide or manage content to the web browser. It can be assisted by technologies like Cascading style sheets(CSS) and scripting languages like JavaScript.

Web browsers receive HTML documents either from the web server or from the direct local storage and render documents to the web browser multimedia.

HTML elements are the building block of HTML. Elements consist of tags, which are written using angular brackets and are needed to manage the content that must be added to the page. Tags contain opening tag, content and closing tag accordingly. Some tags can be self-closing, for example <input>, <img>,
 etc. The tags in HTML are not case sensitive, <IMG> and <img> means the same.

The extension for an html file can be .htm or .html



HTML can embed programs written in a scripting language like JavaScript which affects the behaviour and content of the page, and CSS for positioning the content of the page.

Introduction to Web Designing - CSS

Cascading Style Sheets(CSS) describes how the HTML elements are to be placed in a web page. It saves a lot of work. It can control the layout of multiple pages all at

once. External style sheets can also be added in CSS files.

Attaching a CSS file to a HTML File:

Method -1:

All the required style elements that are needed to apply can be put inside a <style> tag. <style> tag can be placed before or after the body tag in an HTML file.

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>
        body {
                background-color: lightblue;
    }

h1 {
  color: white;
  text-align: center;
}

p {
  font-family: verdana;
  font-size: 20px;
}
    </style>
</head>
```

Method- 2:

The external style sheet can be linked to HTML using a link tag mentioning the path of the CSS file in href.

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
   <link rel="stylesheet" href="style.css">
</head>
```

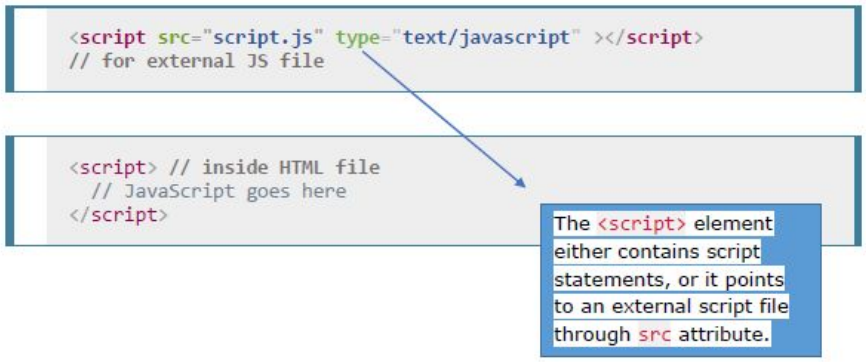
CSS elements can be added either by using the class or id names and can also be done by accessing the elements of html directly.

Introduction to Web Designing - Javascript

JavaScript is a scripting or programming language that allows you to implement complex features on web pages, enables you to dynamically update content, control multimedia, animate images, and pretty much everything else control the behaviour of different elements.

Every time a web page does more than just sit there and display static information for you to look at — displaying timely content updates, interactive maps, animated 2D/3D graphics, scrolling video jukeboxes, etc., languages like JavaScript is probably involved. A very common use of JavaScript is to dynamically modify HTML and CSS to update a user interface, via the Document Object Model API.

A lightweight interpreted programming language. The web browser receives the JavaScript code in its original text form and runs the script from that. From a technical standpoint, most modern JavaScript interpreters actually use a technique called just-in-time compiling to improve performance; the JavaScript source code gets compiled into a faster, binary format while the script is being used, so that it can be run as quickly as possible. However, JavaScript is still considered an interpreted language, since the compilation is handled at run time, rather than ahead of time.
JavaScript is applied to your HTML page in a similar manner to CSS, using the <script> element or a separate external JS file with an extension .js



UI Framework - Bulma

The Bulma CSS framework is a well documented, free and fully open-source CSS solution based on the Flexbox layout. With Bulma, the extensive range of built-in features means faster turnaround and less CSS code writing.

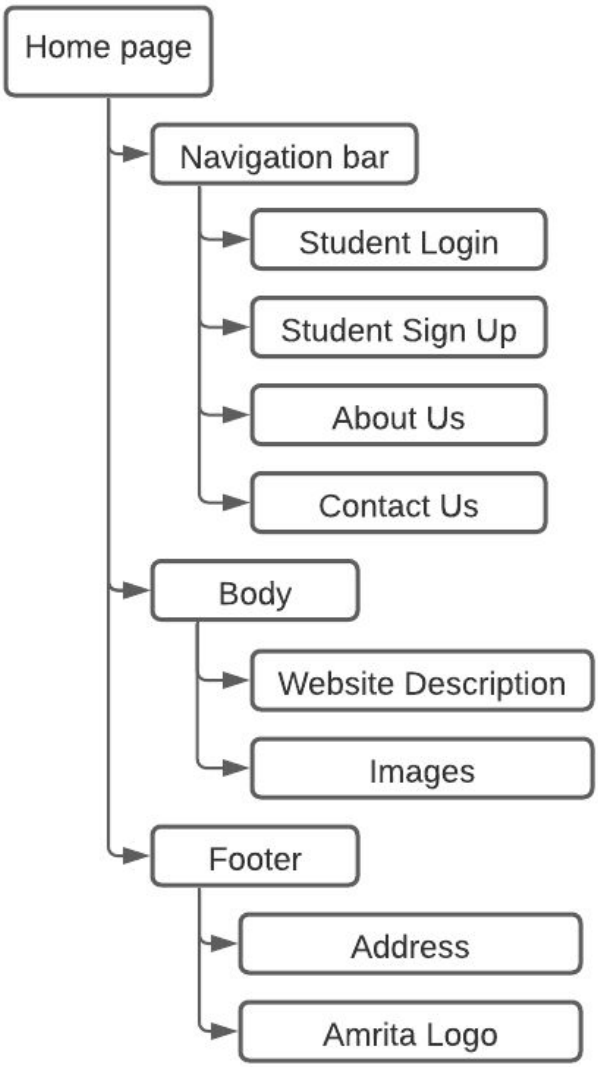
- **Easy to learn** – ease-of-use as a top priority, finding answers to specific questions is no trouble at all, Stack overflow community is extremely Bulma-friendly.
- **Modular** – you can use the pieces you want and leave the rest out.

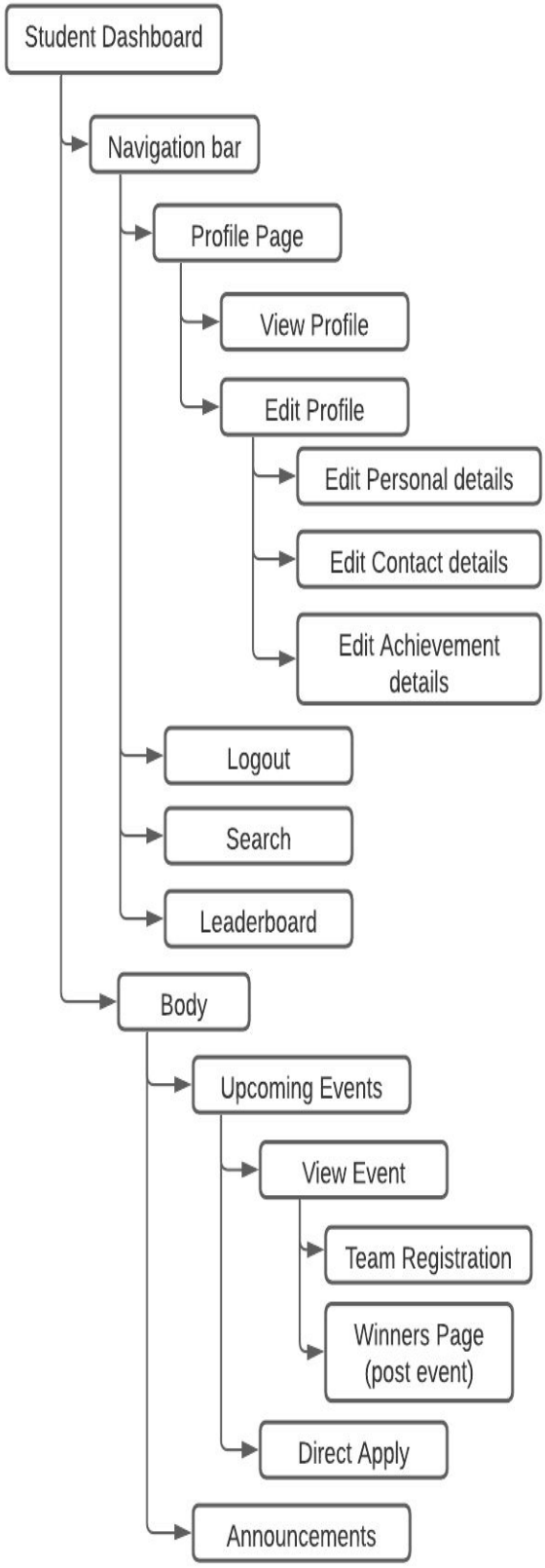
- **CSS only** – the only output is one CSS file (bulma.css), a plug-and-play solution
- **Responsive** – It can be difficult to try to convert a desktop-designed site for use on a mobile device, but with Bulma, it's not really an issue. Bulma is a mobile-first CSS framework, offering optimum site responsiveness.
- **Component-rich** – The components you need to give your site a crisp, functional appearance, including dropdown menus, navigation bars, panels, tabs, and tables
- **Compatible** – Not everyone uses Chrome, so when you're designing a site you want to be sure that it will work well with the full range of available web browsers. Bulma is compatible with many other major browser options.
- **Constantly being improved** – New features are being added on a near-constant basis and newer versions are quickly whittling away at any bugs or other issues users may encounter.

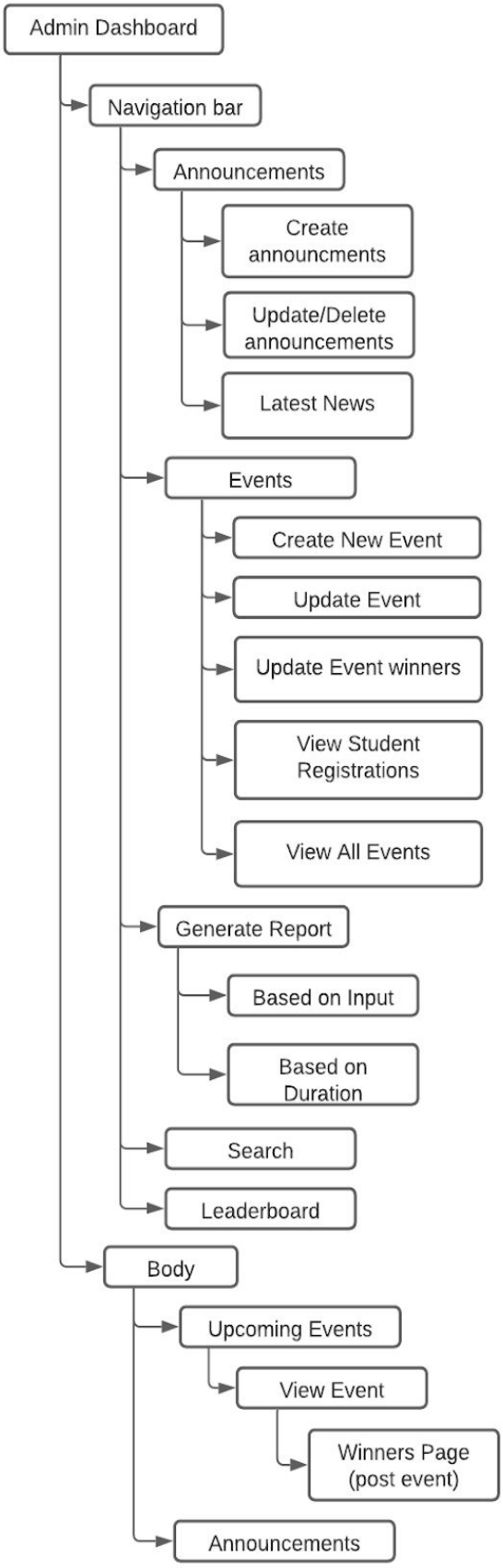
System Requirements and Specifications

- **Client side system requirements and specifications**
    - Modern web browser (Chrome, Mozilla Firefox, etc.)
    - JavaScript should be enabled
- **Server side system requirements and specifications**
    - AWS - RDS
    - Java, JDE, JRE
    - JDBC, Java technologies

Design Diagram




Student Dashboard
Navigation bar
Profile Page
View Profile
Edit Profile
Edit Personal details
Edit Contact details
Edit Achievement details
Logout
Search
Leaderboard
Body
Upcoming Events
View Event
Team Registration
Winners Page (post event)
Direct Apply
Announcements


Admin Dashboard
Navigation bar
Announcements
Create announcments
Update/Delete announcements
Latest News
Events
Create New Event
Update Event
Update Event winners
View Student Registrations
View All Events
Generate Report
Based on Input
Based on Duration
Search
Leaderboard
Body
Upcoming Events
View Event
Winners Page (post event)
Announcements

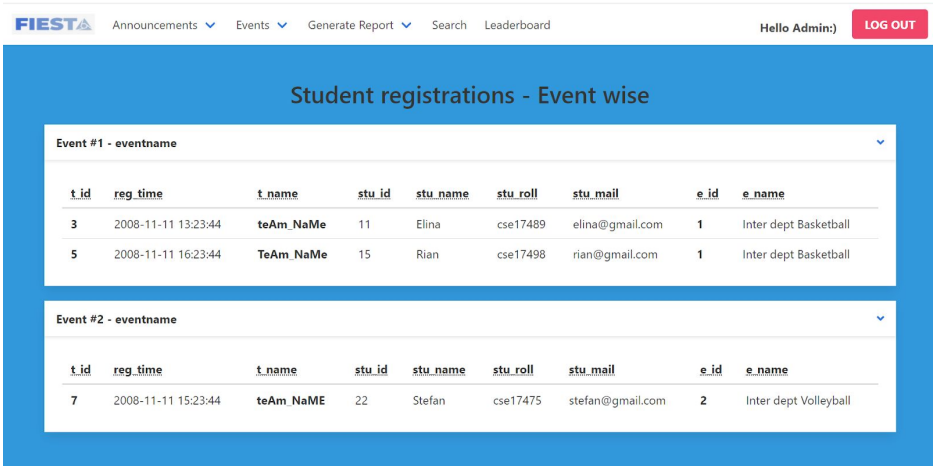
Implementation

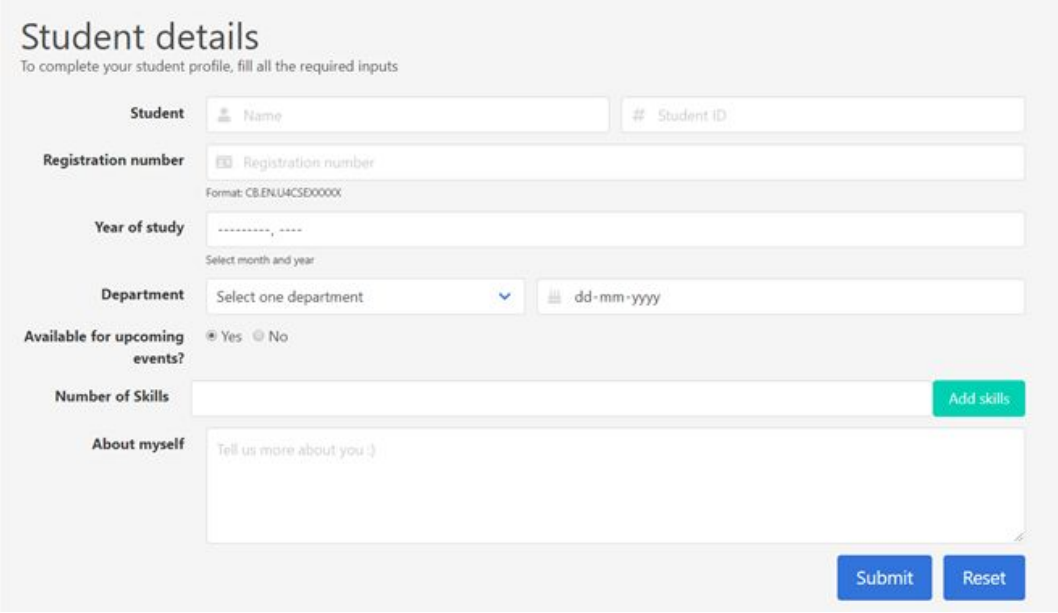
The project Fiesta, an Event Management System is developed keeping the core functionalities of the portal easy and elegant to use by students and admins. The web app is built upon the latest state-of-the-art web technologies. The server code is developed in Java technologies, modularized with appropriate functions and integrated with the frontend. Modularized code ensures effortless debugging and making the development process fast. The SQL database is running on AWS - RDS, which ensures the database is secure, available 24/7.

(A)

Name: **Gayathri E**
Roll No.: **CB.EN.U4CSE17420**
No. of webpages: **6**

A1) Screenshots:





Google drive Folder Link containing remaining screenshots

https://drive.google.com/drive/folders/10hb5uCEYhaUi3Vd2qRGB6LafvPgPR_d3?usp=sharing

A2) Description:

Displaying event details is very much essential in an extra-curricular activities management website for tracking the past and present events. Also, displaying a bunch of details about all the events will make it difficult for the user to read. In order to resolve this issue, all the details of that particular event for which the user is looking is alone displayed which also makes it organized. This is done using a card toggle function using JavaScript. In a similar way, students' details are also displayed event wise using the card toggle function using JavaScript.

Student has to complete his student profile, so that future references will be easier. A keen survey can be taken using the details, he/she has provided (Personal details, Achievement details & Contact details). In the personal details, the user has to enter the skills he/she has. There's a

dynamic control for the number of fields(N) for typing in the skills using the input of number of skills(N). For instance, a user has 4 skills. The user can type 4 and can get 4 fields to enter the skills he has. The number of fields vary from person to person. Having a dynamic control for the number of skills can make the work easier.

A3) Elements used in web page:

Field Name	GUI Control	Validation Rule	Any Specific Feature
Event ID – Event Name	Click on the card title	Card toggles only when the field is clicked	Details of students will be displayed event wise for the respective event ID clicked
Name	Input text	Only alphabets; required	
Student ID	Input number	Only number; required	
Registration number	Input reg. no.	Should follow FORMAT CB.EN.U4CSEXXXXX; required	
Year of study	Choose month & year from the table	Required	
Department	Drop down	Required	
Availability	Radio button	Required	
Date of birth	Date table	Required	
Number of skills	Type number N and then provide details in the N boxes displayed dynamically	User can type only number in this field	Dynamic control of fields for the specific number of skills provided

About myself	Input text	Required	

A4) Code snippets:

JavaScript for dynamic control of number of fields

```
<script type="text/javascript">
    function addFields() {
        debugger;
        var number = document.getElementsByName("no_of_skills")[0].value;
        var container = document.getElementById("create-container");
        container.innerHTML = '';
        for (i = 0; i < number; i++) {
            var skill = document.createTextNode("Skill " + (i + 1));
            var skill = document.createElement("label");
            skill.innerHTML = "Skill " + (i + 1);
            skill.className = "label subtitle is-6";
            container.appendChild(skill);
            var input = document.createElement("input");
            input.type = "text";
            input.name = "name" + i;
            input.className = "input";
            container.appendChild(input);
            container.appendChild(document.createElement("br"));
        }
    }
</script>
```

JavaScript for card toggle

```
<script type="text/javascript">
    document.addEventListener('DOMContentLoaded', function () {
        let cardToggles = document.getElementsByClassName('card-toggle');
        for (let i = 0; i < cardToggles.length; i++) {
            cardToggles[i].addEventListener('click', e => {
                e.currentTarget.parentElement.parentElement.childNodes[3].classList.toggle('is-
hidden');
            });
        }
    });
</script>
```

Inline display of pictures with a snippet of information

```
<div class="columns is-multiline is-centered">
  <div class="column is-8 is-4-widescreen mb-5">
    <div class="block image is-5by4"><img src="mic.jpg" alt=""></div>
    <h4 class="title"><small class="number">1</small> Music</h4>
    <p>Group/Solo Instrumental, Vocal ( Carnatic, Western, Pop), Beatboxing, Sing along events </p>
  </div>
  <div class="column is-8 is-4-widescreen mb-5">
    <div class="block image is-5by4">

      <img src="https://encrypted-tbn0.gstatic.com/images?q=tbn%3AANd9GcRkirS48XaHbJsR-
GJRxOnETIGtv6g0i3Cd3g&usqp=CAU"
        alt="">
    </div>
    <h4 class="title"><small class="number">2</small> Dance</h4>
    <p>Group/Solo Contemporary, Ballet, Hip Hop, Jazz, Ballroom, Folk Dance, Split dance </p>
  </div>
  <div class="column is-8 is-4-widescreen mb-5">
    <div class="block image is-5by4"><img
        src="https://www.mydesignbeauty.com/wp-content/uploads/2016/10/beautiful-oil-paintings-art-
collection-by-mydesignbeauty-1.png"
        alt=""></div>
    <h4 class="title"><small class="number">3</small> Art</h4>
    <p>Solo Oil painting, Wall painting, Digital Arts, Room decor, Glass/Diya painting</p>
  </div>
</div>
```

A5) Testing:

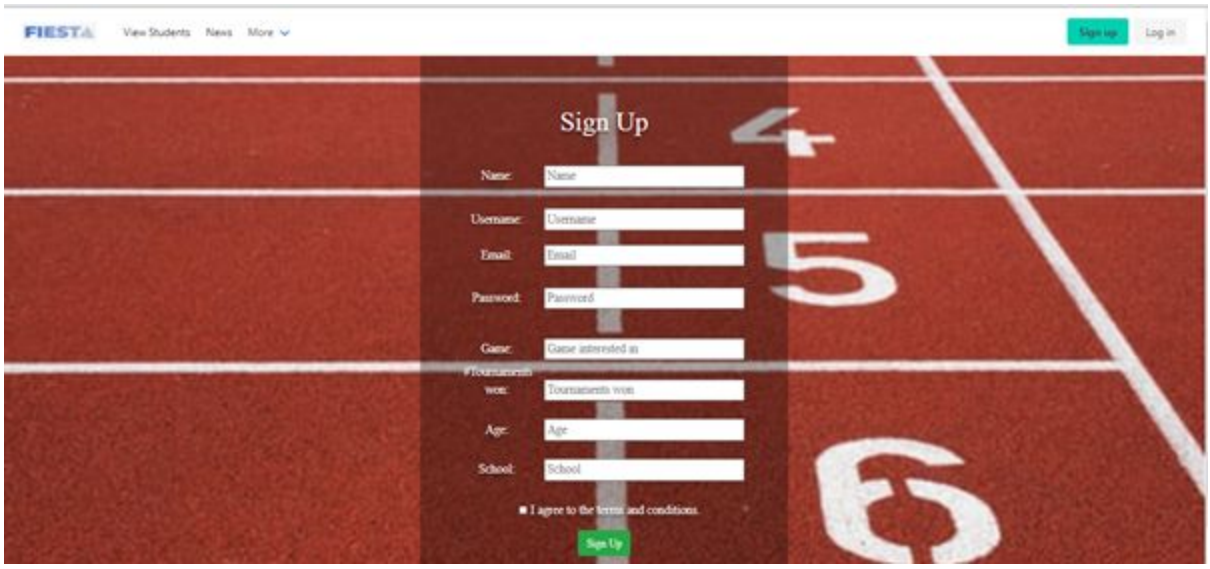
File/Module	Field	Input Given	Success/Failure	Reason for Failure
Personal Details	Student id	ABC	Failure	Accepts only number
Personal Details	Registration number	CB.EN.U4CSE17420	Success	
Personal Details	Registration number	BC.BM.12ABC17490	Failure	Doesn't follow format
Personal Details	Name	NIL	Failure	Mandatory fields required
Personal Details	Number of skills	3	Success	
Personal Details	Number of skills	A	Failure	Only number
Personal Details	About Myself	NIL	Failure	Mandatory fields required

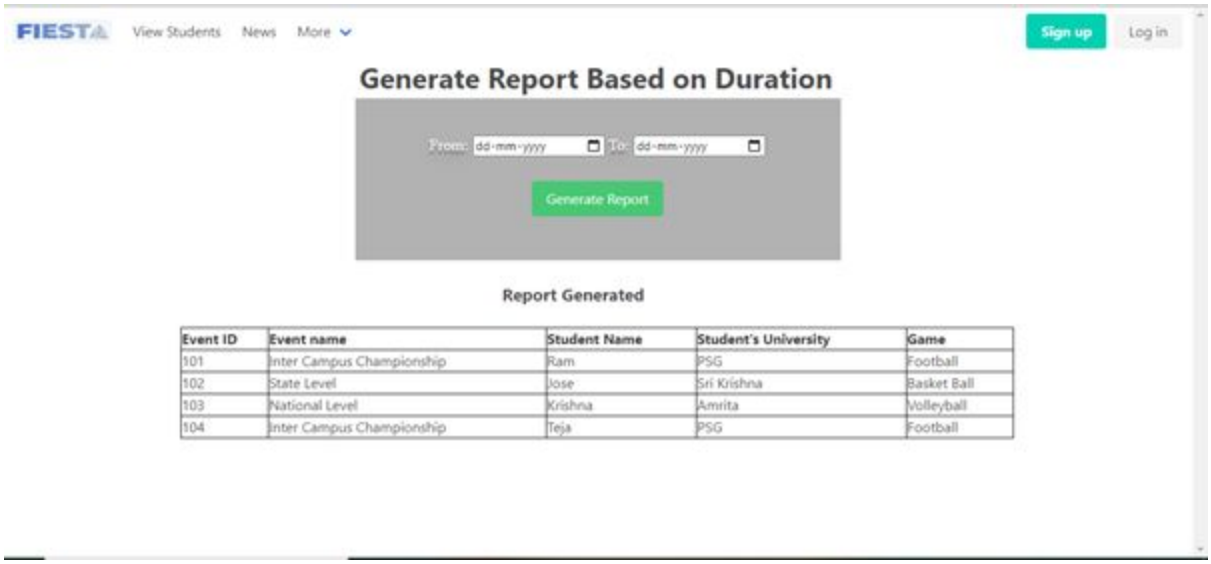
Personal Details	Student id	1234	Success	

(B)

Name: Prathyusha I
Roll No.: CB.EN.U4CSE17424
No. of webpages: 6

B1) Screenshot:





Google drive Folder Link containing remaining screenshots

https://drive.google.com/drive/folders/12-QZR0pkSK6ucq-91f3IIGb8scg3mOJY?usp=sharing

B2) Description:

For every website registration is mandatory for the person to use the website for the required purpose. In the same way for this particular website there is a registration page for students such that the admin can keep track of the students, their profile and the events they take part in.

For easier access of the data, and for reducing the redundancy Database Management System was invented. Using the properties of details of a particular time period can be retrieved and is also very much needed for analysing the data.

B3) Elements used in web page:

Field Name	GUI Control	Validation Rule	Any Specific Feature
Username	Input	Username must not be null.	Serves as a unique identifier.
Password	Password	Passwords must be length 8 and must contain uppercase letters, numbers and special characters as well.	Secures the profile.
Email	Email	Must be in the proper format and should be unique.	Used for retrieving any information in future time.
Event ID	Input	Must not be null	Unique identifier for a particular event.

Check box(for terms and conditions)	Check box	The check box must be checked.	It is the responsibility of the admin to be aware while updating the winners for a particular event and to not create any data discrepancies.

B4) Code snippets:

Validation code for Registration form

```
<script>
    function validateform(){
      var status=false;
      var x= document.forms['for']['Username'].value;
      if(x==""){
          document.getElementById("p1").innerHTML="        Please enter a valid username!" ;
          status=false;
      }
      else{
        document.getElementById("p1").innerHTML="";
        status=true;
      }
      var x=document.forms['for']['Password'].value;
      var pwd=/^(?=.*[0-9])(?=.*[!@#$%^&*])[a-zA-Z0-9!@#$%^&*]{7,15}$/;
      if(!x.match(pwd)){
        document.getElementById("p2").innerHTML="Password must be of 8 characters with one uppercase letter, numbers
and special characters!" ;
        status=false;
      }
      else{
        document.getElementById("p2").innerHTML="";
      }
      return status;
    }
 </script>
```

Validation code for winners update form

```
<script>
  function validateform( ){
    var cb=document.forms['for']['cb'];
  if(!cb.checked){
    alert("Please agree to the terms and confitions!")
  }
}

</script>
```

B5) Testing:

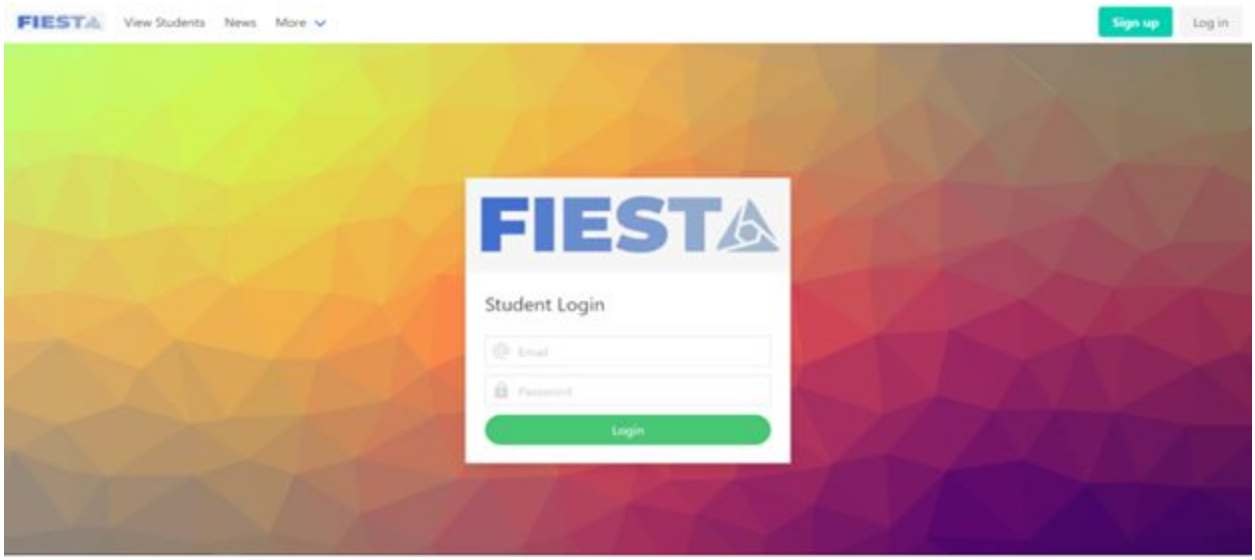
File/Module	Field	Input Given	Success/Failure	Reason for Failure
Registration Form	Username	“ ”	Failure	Username must not be null
Registration form	Username	abraka	Success	
Registration form	Password	prathyu	Failure	Password must be of length 8.
Registration form	Password	prathyusha	Failure	Passwords must contain uppercase letters.
Registration form	Password	Prathyusha	Failure	Password must contain special characters and numbers
Registration form	Password	Prathyush@1234	Success	

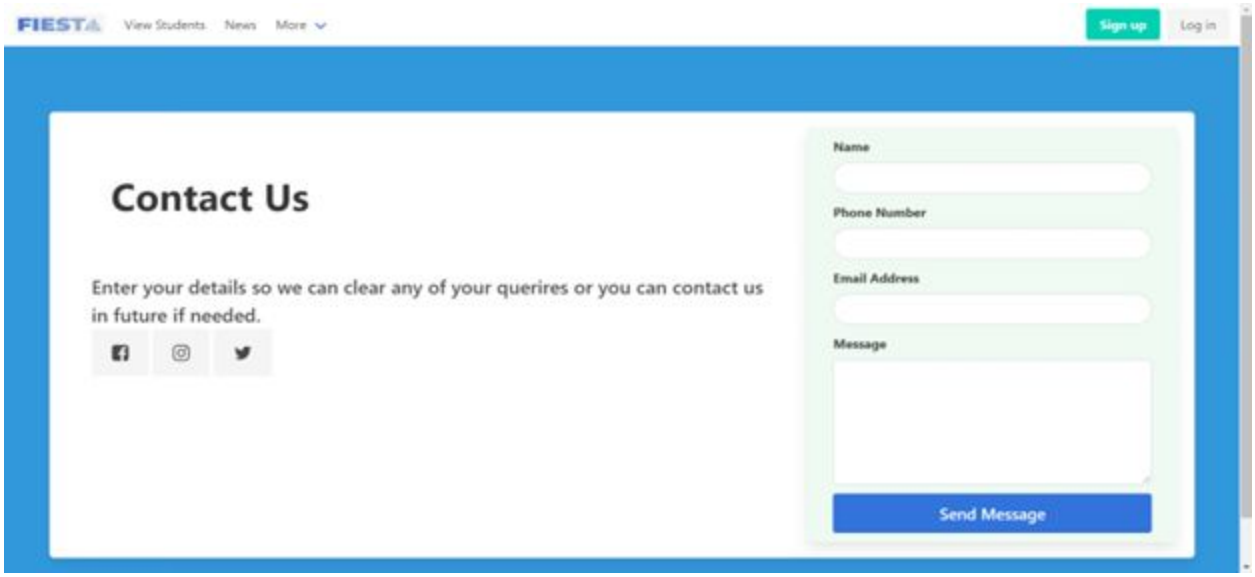
Winners update form	Checkbox	Did not check the checkbox	Failure	Checkbox must be checked.
Winners update form	Checkbox	Checked the checkbox	Success	

(C)

Name: **Neeraj Kumar Reddy P**
Roll No.: **CB.EN.U4CSE17445**
No. of webpages: **5**

C1) Screenshot:





Google drive Folder Link containing remaining screenshots

https://drive.google.com/drive/folders/1CJFo3rUKIZgLcqJGpZSem54FdSxWc7Wd?usp=sharing

C2) Description:

We have a Login form which basically is used to get authenticated credentials which can allow external entities to access a restricted page or form. The login form contains two fields such as a field for the *Email Address* and another field for the *password*. When the login form is submitted its underlying code will check whether the entered credentials are authentic, giving the user access to the restricted page. If a user is not able to provide authentic credentials they will not be able to proceed past the login form. This is very similar to the search form, a login form is basically a type of record form whose insert, update and delete properties have been disabled. Along with the login page is a public Profile page which basically consists of an about row filled data we next split the space into 3 equal boxes to hold 3 different kinds of data, namely contact details, profile holder's picture and skill set. For all these to be implemented I am using bulma a css framework.

There is a contact us page, here its whole content is divided into 2 sections, one section holding the contact us field name and the redirection links of our page. The other section containing text fields where data is to be entered, here 3 fields name, email address and phone number are made mandatory and the message text field is left to be optional. Latest News is a part of the Announcements section which displays the latest events along with its links to access the whole information. Next the two dashboard display pages I used an iframe tag to fit in all the details required into the page.

C3) Elements used in web page:

Field Name	GUI Control	Validation Rule	Any Specific Feature
Email Id	Input text it is a Email	Username should be followed by '@' gmail.com	
Password	Input Text area	Password size between 6-12	
Name	Input text	Only Alphabets Required	
Phone Number	Input Text is Number	Only Number required	
Message	Input Text	Not Mandatory to be entered	

C4) Code snippets:

Student Login page

```
<div class="card-content">
 <div class="media">
 <div class="is-size-4 has-text-centered">Student Login</div>
<p class="control has-icons-left has-icons-right">
<span class="icon is-small is-left">
```

Contact Us page

```
<section class="section has-background-info hero">
           <div class="hero-body">
           <div class="container has-text-centered">
           <div class="box cta is-primary columns is-8 is-variable ">
           <div class=" has-background-white column is-two-thirds has-text-left">
           <h1 class="has-text-weight-bold section title is-1">Contact Us</h1>
           <p class="has-text-weight-semibold is-size-4"></p>
               <a href="https://facebook.com" target="_blank" class="button is-light is-large"><i class="fa
   fa-facebook-square" aria-hidden="true"></i></a>
           <div class="has-background-success-light box cta column is-one-third has-text-left">
<input class="input is-rounded has-text-weight-normal" type="Email" required="">
<button type="submit" class="button is-link is-fullwidth has-text-weight-medium is-medium">
```

Public profile page

```
<section class="hero is-info is-primary">
           <div class="hero-body">
               <div class="container has-text-centered">
               </section>
               <div class="box cta is-light">
           <p class="has-text-centered">
               <section class="container" id="about">
    <div class="has-background-info-light box cta section-heading">
    <h3 class="title is-3">About Michael Phelps</h3>
    <h4 class="subtitle is-5">Most decorated Olympian of all time, with a total of 28 medals.</h4>
    <div class="has-background-white-bis container box">
        <h5 class="subtitle is-6">Short Info:</h5>
        <p class="container has-text-centered">Michael Fred Phelps <strong>(born June 30, 1985)</strong>is an
American former competitive swimmer and the most <strong>successful and most decorated Olympian</strong> of all
time, with a total of <strong>28 medals</strong>. Phelps also holds the all-time records for <strong>[23 Olympic
gold medals]</strong></p>
<div class="columns has-same-height">
    <div class="column is-4">
        <div class="card">
        <div class="card-content">
            <h3 class="title is-4">
<article class="media">
               <div class="media-content">
               <div class="content">
```

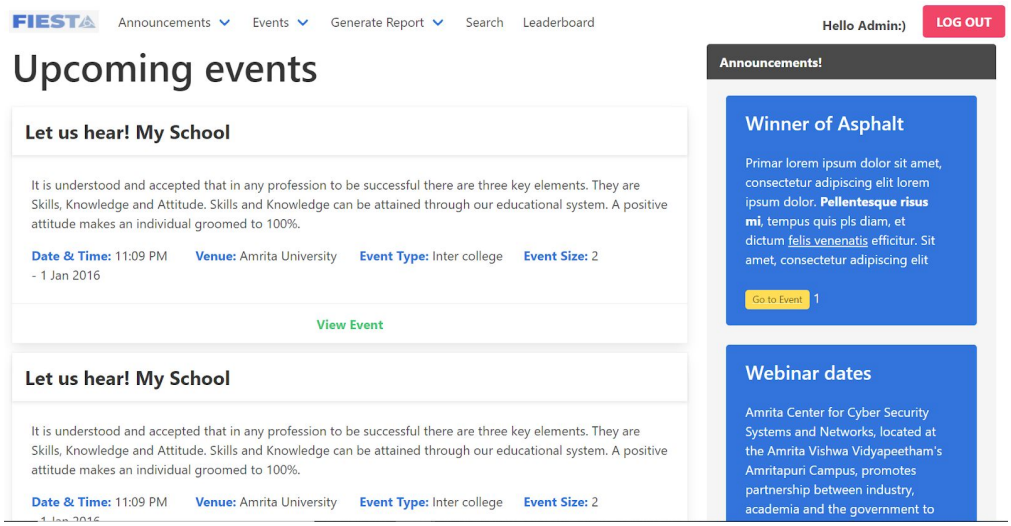
C5) Testing:

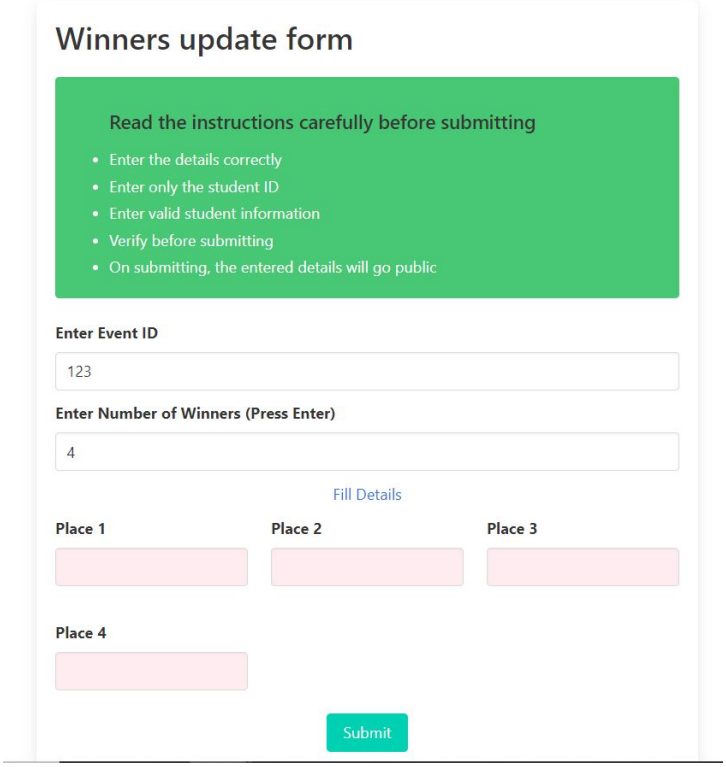
File/Module	Field	Input Given	Success/Failure	Reason for Failure
Student Login	Email id	neeraj	Failure	Not fully completed
Student Login	Email id	neeraj@gmail.com	Success	
Student Login	Password	neeraj	Failure	Size limit not satisfied
Contact US	Name	Nil	Failure	Only Alphabets, and numbers are not accepted
Contact US	Name	Neeraj	Success	
Contact US	Phone Number	9888785543	Success	
Contact Us	Message	hi	Success	
Contact US	Message	hi1233	Success	
Student_Display Dashboard	Search based on tag	sai	Failure	The only accepted tags are @sanjay, @Neeraj, @Gayathri, @Prathyusha, @shilesh
Student_Display Dashboard	Search based on tag	Neeraj	Success	

(D)

Name: Sanjay Tharagesh R. S.
Roll No.: CB.EN.U4CSE17453
No. of webpages: 6

D1) Screenshots:





Google drive Folder Link containing remaining screenshots

https://drive.google.com/drive/folders/1vLAKiJBh0GBhMjLJMgtYtpwGdMAqvT5e?usp=sharing

D2) Description:

The main requirement for students and as well to admins in an event management system is displaying the scheduled upcoming events and competitions. Displaying the event details in an elegant format with important essential details makes sure the user interface and user experience is not cluttered. The upcoming events column is developed using Bulma card-content class. The card also redirects the user to respective event pages and registration form in single click. In some unvoiced circumstances, the events can be cancelled, preponed or postponed, these important modifications will be posted in the announcement section by the Admin. Announcement section is developed using the Bulma message class.

After the successful completion of competitions, the details about the winners and their respective positions can be updated. The form asks for the event ID and number of winners to be declared. Based on the number of winners, respective text fields will be created for the admin to enter the student IDs. All the fields in the form are client side validation using efficient JavaScript functions. The form gets submitted after successful validations.

D3) Elements used in web page:

Field Name	GUI Control	Validation Rule	Any Specific Feature
Event ID – Event Name	Text box	Required	
Number of winners	Text box	Required and accepts only numeric value	Winner ID fields will be created dynamically based on the count
Fill Details	Hyperlink button, onClick()		Dynamic text boxes are created
Place 1, 2	Text box	Required and accepts only numeric value	Prompt title on wrong validation
Announcement ID	Text box	Required and accepts only numeric value	alert() on wrong validation
Update/Delete announcement	Checkbox	Delete – valid announcement ID is check. Update – valid announcement ID and required message(text box)	Alert() on wrong validation, prompt() for confirming deletion.

D4) Code snippets:

JavaScript to validate update winners form

```
<script language="javascript" type="text/javascript">
        let reg = /^\d+$/;
        function addFields() {
            let form = document.getElementById("myForm")
            var number = document.getElementById("count").value;
            if (number == "" || reg.test(number) === false) {
                alert("Enter Valid Number of winners")
                return;
            }
            var container = document.getElementById("winners");
            while (container.hasChildNodes()) {
                container.removeChild(container.lastChild);
            }
            for (i = 0; i < number; i++) {
                let div_ele = document.createElement("div");
                let label = document.createElement("label");
                let input = document.createElement("input");
                label.innerHTML = "Place " + (i + 1);
                div_ele.appendChild(input);
                container.appendChild(div_ele)
            }
        }
        function validate() {
            let id = document.getElementById("event_id").value
            if(id==null || id==undefined || id=="" || reg.test(id) === false) {
                alert("Enter Valid Event ID");
                return false;
            }
            return true;
        }
    </script>
```

JavaScript to validate update / delete announcements

```
<script language="javascript" type="text/javascript">
        let form = document.getElementById("myForm")
        let reg = /^\d+$/
        function validate() {
            ann_id = document.getElementById("announce_id").value
            if(ann_id == null || ann_id == undefined || ann_id == "") {
                alert("Please Enter Announcement ID")
                return false;
            }
            else if(reg.test(ann_id) === false) {
                alert("Enter a Valid Announcement ID")
                return false;
            }
            else if (document.getElementById("update").checked) {
                let txt = document.getElementById("message").value
                if(txt == null || txt == undefined || txt == "") {
                    alert("Fil the message area")
                    return false
                return true
            } else if (document.getElementById("delete").checked) {
                var r = confirm("Are you sure to delete?");
                if (r == true) {
                    return true
                return false

            } else {
                alert("Please select update / delete option")
                return false
            }
        }
    </script>
```

HTML to display upcoming events and announcements

```
<h1 class="title is-1">Upcoming events</h1>
<div class="card mt-3">
    <header class="card-header"> <p class="card-header-title is-size-4">Let us hear! My School</p></header>
    <div class="card-content">
        <div class="content">
            It is understood and accepted that in any profession to be successful there are three key
            elements. They are Skills, Knowledge and Attitude. Skills and Knowledge can be attained through our
            educational system.A positive attitude makes an individual groomed to 100%. <br>
            <div class="columns is-bordered mt-1">
                <div class="column"><span class="has-text-link">Date & Time: </span>11:09 PM - 1 Jan 2016</div>
                <div class="column"><span class="has-text-link">Venue: </span> Amrita University</div>
                <div class="column"><span class="has-text-link">Event Type: </span> Inter college</div>
                <div class="column"><span class="has-text-link">Event Size: </span> 2</div>
            </div>
        </div>
    </div>
    <footer class="card-footer"><a href="../Events/view_event.html" class="card-footer-item has-text-success has-
text-weight-bold">View Event</a></footer>
</div>

<article class="message announce-class">
    <div class="message-header "><p>Announcements!</p></div>
    <div class="message-body">
        <div class="notification is-link">
            <p class="title is-4">Webinar dates</p>
            <p>Amrita Center for Cyber Security Systems and Networks, located at the Amrita Vishwa
                Vidyapeetham's Amritapuri Campus, promotes partnership between industry, academia and the government
to
                foster...</p><br><a href="../Events/view_event.html"><span class="tag is-warning">Go to event</span>
</a>
            <span class="is-primary is-link is-light ann_id ">2</span>
        </div>
    </div>
</article>
```

D5) Testing:

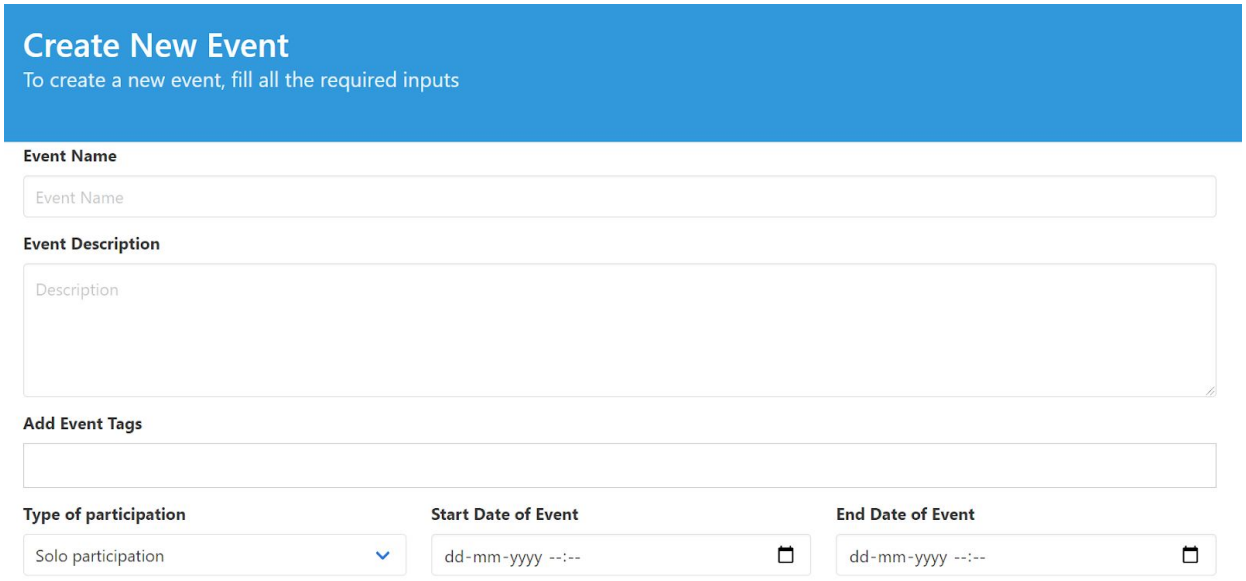
File/Module	Field	Input Given	Success/Failure	Reason for Failure
New Announcements	Announcement message	Empty	Failure	Message cannot be empty
New Announcements	Announcement message	“The inter college football match is postponed to 12-Oct-2020”	Success	
Modify Announcements	Announcement ID	132	Success	

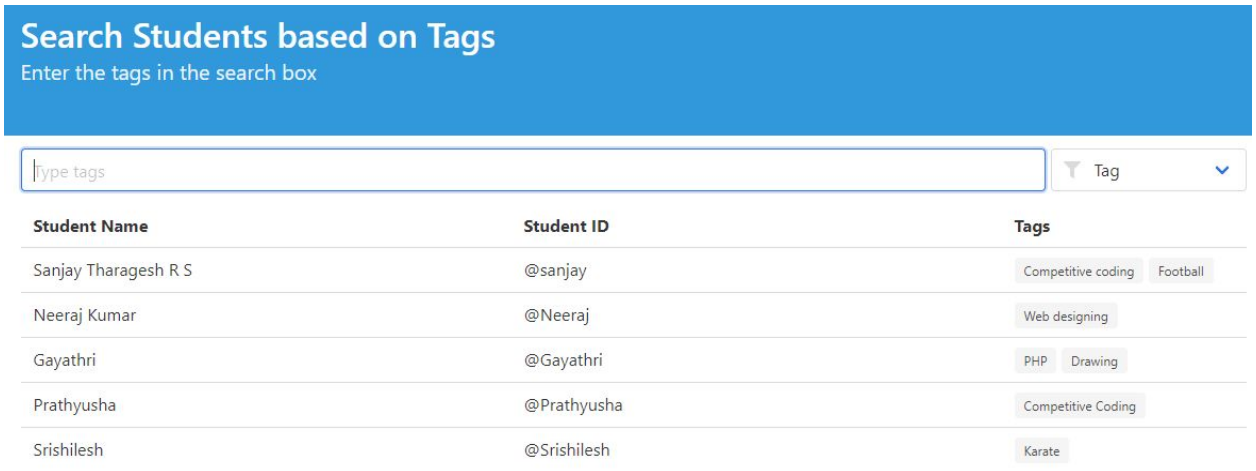
Modify Announcements	Announcement ID	132AP	Failure	Announcement ID should be numeric
New Announcements	Update Announcement (Radio button)	Checked but no announcement body written	Failure	Announcement message should be not empty
New Announcements	Delete Announcement (Radio button)	Checked but no valid announcement ID body	Failure	Announcement ID should be valid
New Announcements	Delete Announcement (Radio button)	Checked and valid announcement ID body	Success	
Update Winners	Event ID	Empty	Failure	Event ID should be Not empty
Update Winners	Event ID	124	Success	
Update Winners	Number of Winners	Empty	Failure	This field is required
Update Winners	Number of Winners	5	Success	
Update Winners	Number of Winners	4sd	Failure	Value should be numeric
Update Winners	Winner (Student) ID	4sd	Failure	Value should be numeric
Update Winners	Winner (Student) ID	443	Success	

(E)

Name: Srishilesh P S
Roll No.: CB.EN.U4CSE17458
No. of webpages: 6

E1) Screenshot:





Google drive Folder Link containing remaining screenshots

https://drive.google.com/drive/folders/1Tk7zEzat0vUaeNfGZbxAjTOryTgAGk1j?usp=sharing

E2) Description:

a) **Admin related pages:**

Admin gets the access to create/update events. Whether the event is intercollege/intra college events, Admin has to fill in all the details regarding the event, organizers, type of event, add necessary tags and other details.

Admin is given access to edit the event details whenever he/she wants.

Admin must enter certain tags related to the events. These tags help us define other related ideas like Leaderboard and search students based on skills.

b) Student Leaderboard:

Student leaderboard is designed to encourage more participants in extracurricular activities. Every student registration in an event is noted, and points are allocated based on the participation or winning the event. Then, the students are ranked based on the ratings. This encourages the students to participate in more such events.

c) Search:

Finding people for the right skills based on their experience of participation in events matters a lot for winning. Our search feature allows Students as well as Admins to filter out students based on the skill tags, Student name and Student ID.

d) Student Registration for an Event(Solo/Team):

Registration is a hefty process for an event. Our solution provides a much simpler way to register.

For solo event, one click of register button registers them to the event

For team events, the number of participants on being set by the admin, the team leader has to enter the student ID/Name in the fields to register.

E3) Elements used in web page:

Field Name	GUI Control	Validation Rule	Any Specific Feature
Event ID/ Name / Description	If new, Write the name or else choose from dropdown	Only letters - set maxlength	
Add Tags	Write down tags, automatically enters unique tags, enhanced UI	Validation of similar tags	Display a unique set of tags. The tags are space separated. HIghlights automatically on next

			word Uses jQuery to add tags
Start/End date of event	Choose date and time of event from dropdown calendar	Automatic validation	
Email/Phone number	Checks continuously if the pattern is matched	Automatic client side validation of respective types	
Registration link	Field for entering URL	Validates if the URL is valid	For 3rd party registration apart from portal registration
Leaderboard table	Displays the student leaderboard containing skills, ID and Name	No validation required	Sorts the tuples based on scores
Search feature	Filtering options of students skills, Name and ID	No validation required	Pattern matching of tags done continuously
Team registration fields	Creates N number of fields based on Admin specs	Select from dropdown	Dynamic pattern matching from datalist. Based on number of fields, N number of fields are displayed

E4) Code snippets:

JavaScript for validating Create/Update events

```
<script>
    function validSubmission() {
      var email = document.getElementsByName('event_organizer_email')
      var phone = document.getElementsByName('event_organizer_phone')
      if (validateEmailAddress(email)) {
        if (validPhone(phone)) {
          alert('Form Submitted')
        }
      }
    }

    function validateEmailAddress(email) {
      var re =
        /^(([^<>()[\]\\.,;:\s@\"]+(\.[^<>()[\]\\.,;:\s@\"]+)*)|(\".+\"))@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-
9]{1,3}\])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))$/
      if (!email.value.match(re)) {
        alert("Invalid Email address")
        email.focus()
        return false
      }
      return true
    }

    function validatePhone(phone) {
      var re = /^([0|\+[0-9]{1,5})?([7-9][0-9]{9})$/
      if (!phone.value.match(re)) {
        alert("Invalid Phone number")
        phone.focus()
        return false
      }
      return true
    }
  </script>
```

JavaScript for dynamic sorting Leaderboard based on scores

```
<script>
        sortTable();
        function sortTable() {
            var table, rows, switching, i, x, y, shouldSwitch;
            table = document.getElementById("myTable");
            switching = true;
            /*Make a loop that will continue until
            no switching has been done:*/
            while (switching) {
                //start by saying: no switching is done:
                switching = false;
                rows = table.rows;
                /*Loop through all table rows (except the
                first, which contains table headers):*/
                for (i = 1; i < (rows.length - 1); i++) {
                    // start by saying there should be no switching:
                    shouldSwitch=false;
                    /*Get the two elements you want to compare,
                    one from current row and one from the next: */
                    x = rows[i].getElementsByTagName("td")[2];
                    y = rows[i + 1].getElementsByTagName("td")[2];
                    //check if the two rows should switch place:
                    if (Number(x.innerHTML) < Number(y.innerHTML)) {
                        //if so, mark as a switch and break the loop:
                        shouldSwitch = true;
                        break;
                    }
                }
                if (shouldSwitch) {
                    /*If a switch has been marked, make the switch
                    and mark that a switch has been done:*/
                    rows[i].parentNode.insertBefore(rows[i + 1], rows[i]);
                    switching = true;
                }
            }
        }
    </script>
```

JavaScript for Filtering based on skills, ID and Name:

```
<script>
        function myFunction() {
            var input, filter, table, tr, td, i, txtValue, filter_option, filter;
            var f = 2;
            filter_option = document.getElementById("filteroption");
            filter = filter_option.options[filter_option.selectedIndex].value;
            if (filter == 0)
                f = 0
            else if (filter == 1)
                f = 1;
            else
                f = 2;
            input = document.getElementById("myInput");
            filter = input.value.toUpperCase();
            table = document.getElementById("myTable");
            tr = table.getElementsByTagName("tr");
            for (i = 0; i < tr.length; i++) {
                td = tr[i].getElementsByTagName("td")[f];
                if (td) {
                    txtValue = td.textContent || td.innerText;
                    if (txtValue.toUpperCase().indexOf(filter) > -1) {
                        tr[i].style.display = "";
                    } else {
                        tr[i].style.display = "none";
                    }
                }
            }
        }
    </script>
```

HTML & JavaScript for Dynamic fields rendering for Team registration:

```
<div class="container">
    <div class="field">
        <label class="label">Number of Participants (Remove this - Use backend)</label>
        <div class="control">
          <input name="no_of_participants" class="input is-" type="number" id="no_of_participants" required>
          <button class="button is-primary" name="participant" onclick="addFields();">Add Names</button>
          <h4 class="title is-4">1st participant is the Team leader</h4>
          <form action="" method="">
            <div class="field" id="create-container">
            </div>
            <input type="submit" class="button is-primary is-centered" value="Submit">
            <input type="reset" class="button is-primary is-centered" value="Reset">
          </form>
        </div>
      </div>
</div>

<script>
      function addFields() {
        debugger;
        var number = document.getElementsByName("no_of_participants")[0].value;
        var container = document.getElementById("create-container");
        container.innerHTML = '';
        for (i = 0; i < number; i++) {
                var participant = document.createTextNode("Participant " + (i + 1));
                var participant = document.createElement("label");
                participant.innerHTML = "Participant "+(i + 1);
                participant.className = "subtitle is-4";
                container.appendChild(participant);
                var input = document.createElement("input");
                input.type = "text";
                input.name = "name" + i;
                input.className = "input";
                container.appendChild(input);
                container.appendChild(document.createElement("br"));
            }
        }
  </script>
```

E5) Testing:

File/Module	Field	Input Given	Success/Failure	Reason for Failure
Admin Create/Update Event	Event Name	Anokha Fest	Success	
Admin Create/Update Event	Max No of participants	Hundred	Failure	Must be Natural number
Admin Create/Update Event	Max No of participants	1000	Success	
Admin Create/Update Event	Tags	123	Failure	Must be alphabets, separated by spaces
Student Team Registration	Name/ID	123 - Srishilesh	Success	Choosing from dropdown
Search based on tags	Search field	No validation required	Success	Only pattern matching

Admin Create/Update Event	Organizer email	1234@.com	Failure	Must match email format
Admin Create/Update Event	Email	asdf@gmail.com	Success	
Admin Create/Update Event	Phone	12345	Failure	Must be of length 10
Student team registration	Number of participants	100	Success	

Evaluation Sheet

Roll No	Technology	Max Marks	Marks Awarded
CB.EN.U4CSE17420	HTML CSS JS Viva	10 10 10 10	
CB.EN.U4CSE17424	HTML CSS JS Viva	10 10 10 10	
CB.EN.U4CSE17445	HTML CSS JS Viva	10 10 10 10	
CB.EN.U4CSE17453	HTML CSS JS	10 10 10	

	Viva	10	
CB.EN.U4CSE17458	HTML CSS JS Viva	10 10 10 10	
	Project Documentation	10	
	Total	50	

TERM 2

Overview on XML

XML (eXtensible Markup Language) is a simple and a very flexible markup language. It improves on the HTML approach and makes the web a better place to do business, to learn, and to have fun. HTML is a great technology and it has changed the world. However, a great deal of useful information is lost when data is converted into HTML. If this information can be preserved then it can be used to build a whole New World of more information. XML is all about preserving useful information on the web. XML is a markup language that can run on any platform, operating system, or environment and is designed to provide developers with a mechanism to describe their content better. Originally, it was designed for publishing projects, but later it has been developed to make exchange of data on the web much easier and efficient. It was easy to write XML but then the problem arose how to validate according to the rules. The first form of validation was possible using syntax's very similar to XML called DTD.

DTDs are markup language rulebooks that define what markup elements can be used to describe a document. If one wants to create one's own tags then we first need to define the tag in the DTD. XML is a flexible framework to create one's own customized markup languages. All XML-based languages will share the same look and feel. They will all share a common basic syntax. After this, the developer is free to build his/her own diverse markup languages XML requires a browser which has a parser which can understand the syntaxes of XML.

E.g: Internet Explorer. This browser has the XML built in parser. There are two types of parsers, one which only checks for the well formed document and the second one is known as validating parser. The validating parser will also check whether an XML document is well formed as well as validation of DTDs. The latest browser by Netscape i.e. Netscape supports xml. XML files are saved with . XML extension. When there is complex information to be evaluated and executed XML Schemas are used.

XML Schemas are better than DTDs by allowing you to associate rich data types with elements, and eventually with attributes as well. In a DTD, element content is limited to strings and a few other primitive data types. XML Schema supports a wide range of rich data types, such as integers, floating point, numbers, dates, and times. XML Schema also includes support for other features, such as an open content model and namespace integration. The following is a list of the major benefits offered by XML Schema compared to DTDs:

1. XML Schemas are based on XML, not some specialized syntax.
2. XML can be parsed and manipulated just like other XML documents.
3. XML Schemas support a variety of data types. (int, float, Boolean, date etc.)
4.XML Schema presents an open-ended data model, which allows you to extend vocabularies and establish inheritance relationships between elements without invalidating documents.

5. XML Schemas support namespace integration, which allows you to associate individual nodes of a document with type declarations in a schema.
6. XML Schemas support attribute groups, which allow you to logically combine attributes.

In short we can say XML Schema is a much-advanced form of DTDs. Now, when executed alone, the XML documents will show an output as it was written. XML documents are not formatted by default as HTML documents are. CSS is one of those stylesheets. Thus CSS are used for formatting XML documents. (CSS) is a relatively simple tool that allows the developer to assign styles to HTML elements. CSS duplicates formatting built into HTML. It provides web developers with access to a large variety of formatting properties such as margins, line-height, word spacing, and many more.

One XML document can be formatted in many different ways just by changing the style sheet. Different style sheets can be designed for different purposes. CSS actually works better with XML than with HTML because HTML is burdened with backward compatibility between CSS and HTML tags. CSS is easy to learn and style sheets can be included directly in xml documents or can be saved as standalone text files. The only drawback is the style sheet is browser related and hence the developer needs to take care while developing it.

A CSS style is required in order to display XML documents in a presentable form, it is necessary to have a mechanism to describe how the document should be displayed. One of these mechanisms that we have seen is CSS. Apart from this, the W3C has proposed a style-sheet language for XML, called Extensible Stylesheet Language (XSL). XSL is a style sheet language of XML, and XSL is far more sophisticated than CSS. XSL was developed from a proposal forwarded by a group of editors representing Microsoft, Inso, ArborText, and the University of Edinburgh, along with James Clark. XSL makes XML more powerful which enables conditional formatting and application of programming to an extent.

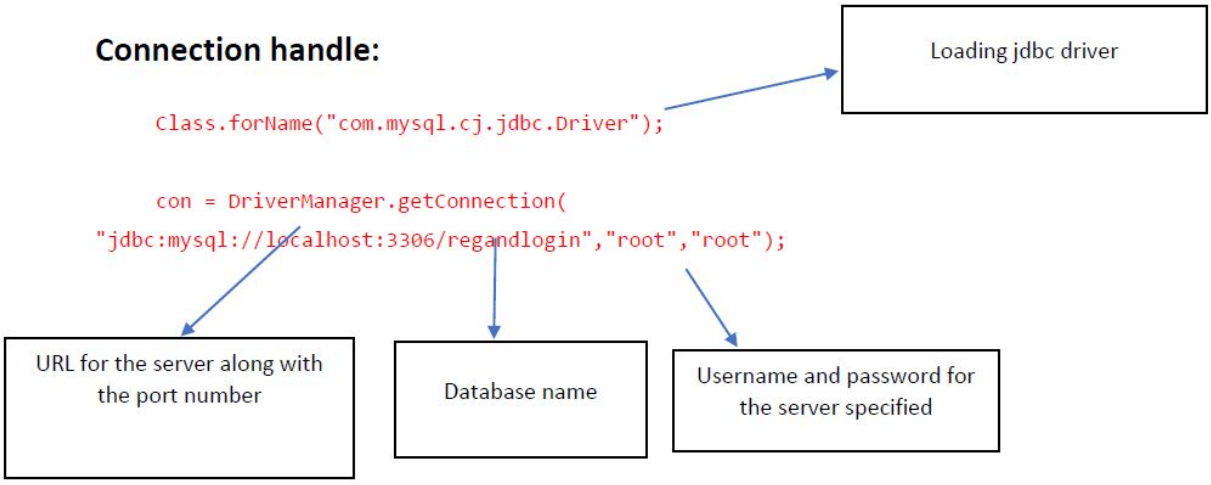
Document Object Model (DOM) is also an important aspect of XML. DOM is used to create a tree structure of a well-formed xml document. DOM facilitates adding of objects to tree structure. DOM is the means by which users can access and manipulate XML documents. With DOM, XML promises an application-independent programming language. XML is one of the popular languages used on the web for data interchange. The DOM output can be viewed in a web browser. The web browser has its own object hierarchy in which the Document object is at top, that is why it is known as Document object model.

In the Object oriented paradigm, a common DOM API is used to allow programmers to write one set of code that can interpret documents running on any application or platform. DOM is a platform- and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents.

Let's take an example of the company, which is having centralized employee records at X office maintained in XML format. This company has four more offices in the metropolitan area. A new employee can join any of the branches. The responsibility of updating employee records is of the operator. Operator has to update, if any new designation is added to the organization. For updating, the first requirement is interface, which is provided by DOM to access the centralized employee records from any of the offices

JDBC

JDBC(Java Database connection), as the name itself justifies it is an API(Application Programming Interface) in Java language which specifies how a client can access the database. It is a java based data access technology used for java database connectivity. It provides methods to query and update data in database, and update data in database.



PreparedStatement:

A Java JDBC PreparedStatement is a special kind of Java JDBC statement object with some useful additional features. Remember, you need a Statement in order to execute either a query or an update .Java JDBC PreparedStatement can be used instead of a Statement and benefit from the features of the PreparedStatement.

The Java JDBC PreparedStatement primary features are:

- Easy to insert parameters into the SQL statement.
- Easy to reuse the PreparedStatement with new parameter values.
- May increase performance of executed statements.
- Enables easier batch updates.

Example:

```
String sql = "update people set firstname=? , lastname=? where id=?";

PreparedStatement preparedStatement =
        connection.prepareStatement(sql);

preparedStatement.setString(1, "Gary");
preparedStatement.setString(2, "Larson");
preparedStatement.setLong  (3, 123);

int rowsAffected = preparedStatement.executeUpdate();
```

As it can be seen in the above example that the rows in the table are updated, if needed to be updated in later times, the same can be done using the same preparedstatement. Not the whole row in the table but one tuple in a row can be updated or retrieved using preparedstatement.

Entity Relationship Diagram

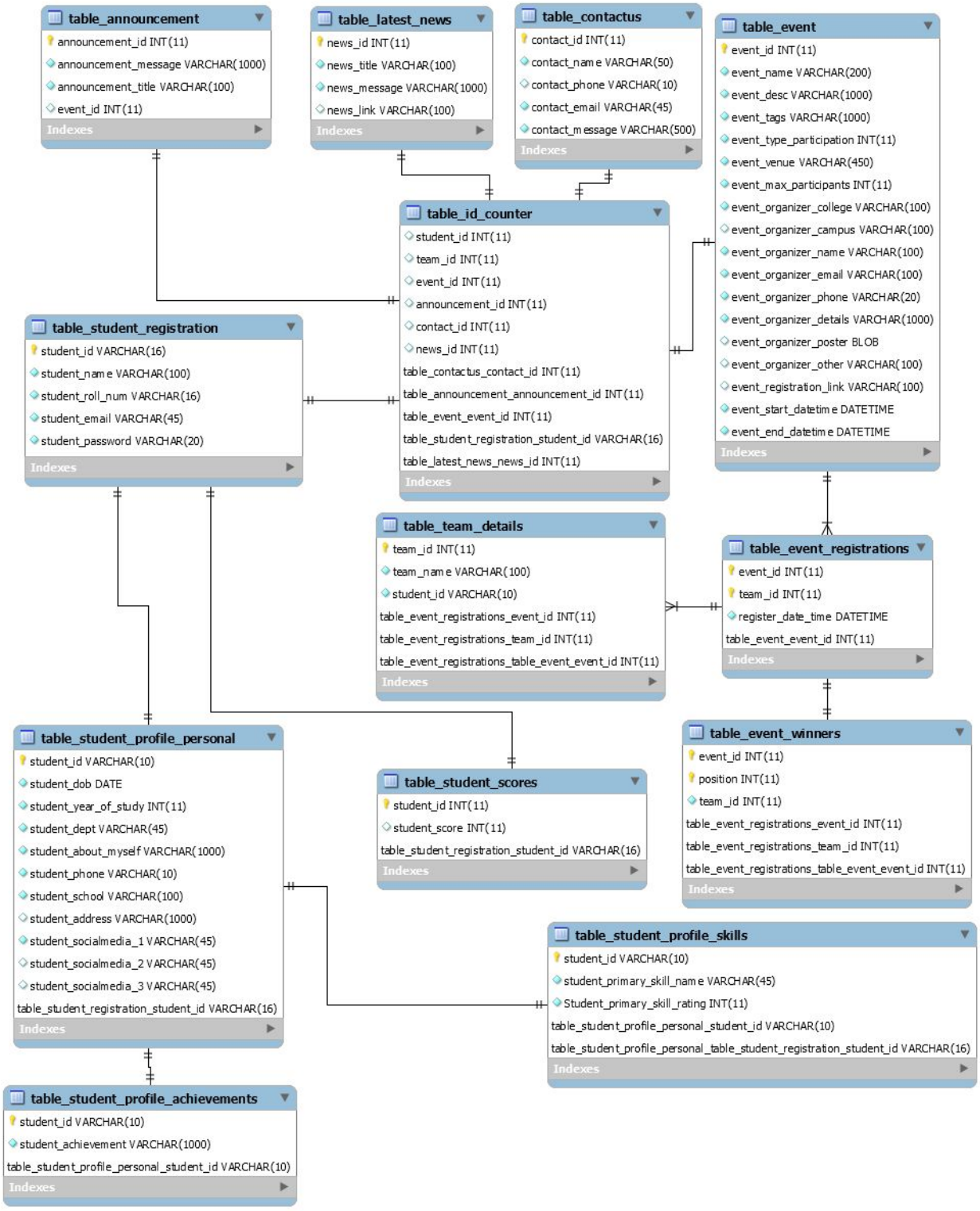


Table Specifications

S. No	Field	Type	Purpose
1	student_id (P)	VARCHAR(16)	To identify the students registered in the portal (The format is Number)
2	student_name	VARCHAR(100)	To identify the name of the student
3	student_roll_num	VARCHAR(16)	College Roll Number of the student
4	student_email	VARCHAR(45)	Student email address
5	student_password	VARCHAR(20)	Password set based on certain constraints
6	contact_id (P)	INT	To identify a contact message in the portal
7	contact_name	VARCHAR(50)	Name of the person who has raised the query
8	contact_phone	VARCHAR(10)	Phone number of the person
9	contact_email	VARCHAR(45)	Email address of the person who has raised the queryai
10	contact_message	VARCHAR(500)	Message raised as the query
11	announcement_id (P)	INT	To identify announcements based on the unique address
12	announcement_title	VARCHAR(100)	To identify the title of the announcements
13	Announcement_message	VARCHAR(1000)	The announcement message to be posted
14	event_id	INT	To identify the event based on unique ID
15	event_name	VARCHAR(200)	To identify the name of the event
16	event_desc	VARCHAR(1000)	The event description
17	event_tags	VARCHAR(1000)	The tags related to the events are separated by commas, which is later used for generating reports
18	event_type_participation (solo - 0, team - 1)	INT	The type of event which requires solo or team participation
19	event_start_datetime	DATETIME	The start date and time of the event
20	event_end_datetime	DATETIME	The end date and time of the event
21	event_venue	VARCHAR(450)	The venue of the event
22	event_max_participants	INT	Maximum number of participants that the event allows
23	event_registration_link	VARCHAR(100)	External registration form if any
24	event_organizer_college	VARCHAR(100)	The Organizing college (if intracollege event)
25	event_organizer_campus	VARCHAR(100)	The Organizing campus(if intracollege event)
26	event_organizer_other	VARCHAR(100)	The details of the organizer

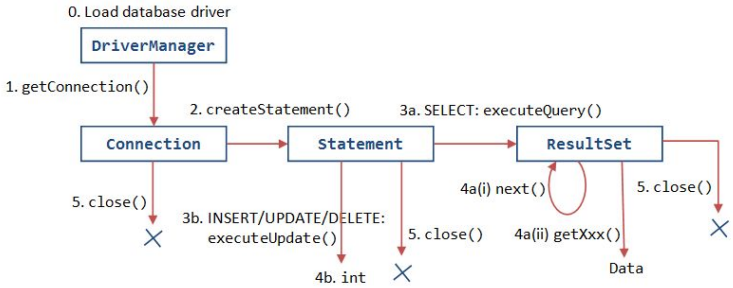
27	event_organizer_name	VARCHAR(100)	The organizer name
28	event_organizer_email	VARCHAR(100)	Organizers email address
29	event_organizer_phone	VARCHAR(20)	Organizers phone number
30	event_organizer_details	VARCHAR(1000)	Details related to the organizer
31	event_organizer_poster	BLOB	Poster related to the event, if any
32	team_id (P)	INT	To identify unique teams
33	team_name (If solo - replace with Student name)	VARCHAR(100)	To identify the name of the teams
34	registration_datetime	DATETIME	To identify the date and time of team registration for any particular event
35	position	INT	To identify the winner position for any particular event
37	student_phone	VARCHAR(10)	Student mobile number
38	student_dob	DATE	Date of Birth of the student
39	student_school	VARCHAR(100)	School name of the student
40	student_year_of_study	INT	Current year of study of the student
41	student_dept	VARCHAR(45)	Department name of the student
42	student_address	VARCHAR(1000)	Address of the student in text format
43	student_about_myself	VARCHAR(1000)	Brief details of the student in text format
44	student_socialmedia_1	VARCHAR(45)	Social Media handle 1 of the student
45	student_socialmedia_2	VARCHAR(45)	Social Media handle 2 of the student
46	student_socialmedia_3	VARCHAR(45)	Social Media handle 3 of the student
47	student_primary_skill_name	VARCHAR(45)	Name of the student's primary skill
48	student_primary_skill_rating	INT	Rating of the student's skill in scale of 1 to 5 (only integers)
49	Student_achievement	VARCHAR(1000)	Student achievement in text format
50	News_id (P)	INT	To identify the News registered in the portal
51	News_title	VARCHAR(100)	Title of the news
52	News_message	VARCHAR(1000)	Message to the respective news
53	News_link	VARCHAR(100)	URL to the external news article
54	student_score	INT	Total score obtained by the student so far

Steps for JDBC connectivity

There are 5 steps to connect any java application with the database using Java Database Connectivity (JDBC).

JDBC: API for the Java programming language that defines how a client may access a database.
Steps involved are as follows:

1. Register the Driver class
2. Create connection
3. Create statement
4. Execute queries
5. Close connection



1) Register the driver class

The forName() method of Class class is used to register the driver class. This method is used to dynamically load the driver class.

```
public static void forName(String className)throws ClassNotFoundException
```

->For loading oracle driver to establish database connection,

```
Class.forName("oracle.jdbc.driver.OracleDriver");
```

---

2) Create the connection object

The getConnection() method of DriverManager class is used to establish connection with the database.

```
public static Connection getConnection(String url,String name,String password)
throws SQLException
```

---

3) Create the Statement object

The createStatement() method of the Connection interface is used to create statements. The object of statement is responsible to execute queries with the database.

```
public Statement createStatement()throws SQLException
```

Statement stmt=con.createStatement();

---

4) Execute the query

The executeQuery() method of Statement interface is used to execute queries to the database. This method returns the object of ResultSet that can be used to get all the records of a table.

```
public ResultSet executeQuery(String sql)throws SQLException
```

Example query

```
1. ResultSet rs=stmt.executeQuery("select * from emp");
2. while(rs.next()){
3. System.out.println(rs.getInt(1)+" "+rs.getString(2));
4. }
```

---

5) Close the connection object

By closing the connection object statement and ResultSet will be closed automatically. The close() method of Connection interface is used to close the connection.

```
public void close()throws SQLException
```

To close the connection

con.close();

```xml
<?xml version="1.0" encoding="UTF-8"?>
<root>
    <student_id>0</student_id>
    <team_id>0</team_id>
    <event_id>0</event_id>
    <announcement_id>0</announcement_id>
    <contact_id>0</contact_id>
    <news_id>0</news_id>
</root>
```

```xml
<root>
    <achievement>
        <student_id> 1 </student_id>
        <student_achievement> Awarded as the best stand-up comedian
and got second place in intra-basketball event</student_achievement>
    </achievement>
 </root>
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<root>
    <event_id>0</event_id>
    <team_id>0</team_id>
 <registration_datetime>0</registration_datetime>
</root>
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<root>
    <News_id>0</News_id>
    <News_title>0</News_title>
    <News_message>0</News_message>
    <News_link>0</News_link>
    <News_id>1</News_id>
    <News_title>1</News_title>
    <News_message>1</News_message>
    <News_link>1</News_link>
</root>
```

```xml
<student>
<row>
   <id>1</id>
   <name>Gayu</name>
   <rollnumber>20</rollnumber>
   <gmail>gayu20@gmail.com</gmail>
   <password>Egayu#120</password>
</row>
<row>
   <id>2</id>
   <name>Prathyu</name>
   <rollnumber>30</rollnumber>
   <gmail>prath20@gmail.com</gmail>
   <password>prafiesta</password>
</row>
</student>
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<root>
    <profile-personal>
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<root>
    <student_id>0</student_id>
    <student_score>0</student_score>
</root>
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<team>
<row>
   <id>210</id>
   <name>Igniters</name>
   <student_id>1</student_id>
</row>
</team>
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<skillset>
<student>
    <student_id>1</student_id>
    <primary_skill>Badminton</primary_skill>
    <ps_rating>2.5</ps_rating>
</student>
<student>
    <student_id>2</student_id>
    <primary_skill>Football</primary_skill>
    <ps_rating>4.5</ps_rating>
</student>
</skillset>
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<contactus>
    <row>
        <contact_id>1</contact_id>
        <contact_name>Krishna Teja</contact_name>
        <contact_phone>9878987656</contact_phone>
        <contact_email>teja@gmail.com</contact_email>
        <contact_message>Registration form is nopt
working.</contact_message>
    </row>
</contactus>
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<announcements>
    <row>
        <announcement_id>1</announcement_id>
        <announcement_title>Event Postponed</announcement_title>
        <announcement_message>The FIFA gaming event that was supposed
to be held on September 30 has been postponed to October
29th.</announcement_message>
        <event_id>12</event_id>
    </row>
</announcements>
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
<event_winners>
    <row>
        <event_id>12</event_id>
        <team_id>1</team_id>
        <position>1</position>
```

```
        <student_id>1</student_id>
        <student_dob>2001-01-02</student_dob>
        <student_year_of_study> 1</student_year_of_study>
        <student_dept> Computer Science</student_dept>
        <student_about_myself> An athlete who competes in many sports
that involve physical strength, speed or endurance.
</student_about_myself>
        <student_phone> 9498058096 </student_phone>
        <student_school> Amrita School of Eng.</student_school>
        <student_address> No.123, XYZ Flats, ABC
City.</student_address>
        <student_socialmedia>
            <student_socialmedia_1> https://twitter.com/dhanushkraja
</student_socialmedia_1>
            <student_socialmedia_2>
https://www.instagram.com/dhanushkraja </student_socialmedia_2>
            <student_socialmedia_3>
https://www.facebook.com/DhanushKRaja </student_socialmedia_3>
        </student_socialmedia>
    </profile-personal>
</root>
```

```
    </row>
</event_winners>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
    <event>
        <event_id>1</event_id>
        <event_name>ICPC</event_name>
        <event_desc>5 coding questions in 2 hrs</event_desc>
        <event_tags>
            <tag id="1" displayName="coding">coding</tag>
        </event_tags>
        <event_type_participation>Team</event_type_participation>
        <event_timing>
<event_start_datetime>2020-04-01T10:00:00</event_start_datetime>
<event_end_datetime>2020-04-01T12:00:00</event_end_datetime>
        </event_timing>
        <event_venue>AB2 CP Lab2</event_venue>
        <event_max_participants>100</event_max_participants>
        <event_registration_link></event_registration_link>
        <event_organizer>
            <event_organizer_intracollege>
<event_organizer_college>Amrita</event_organizer_college>
<event_organizer_campus>Coimbatore</event_organizer_campus>
            </event_organizer_intracollege>
            <event_organizer_other></event_organizer_other>
            <event_organizer_name>Dr.Abc</event_organizer_name>
<event_organizer_email>abc@gmail.com</event_organizer_email>
            <event_organizer_phone>1234567890</event_organizer_phone>
<event_organizer_details>asdfqwer</event_organizer_details>
            <event_organizer_poster></event_organizer_poster>
        </event_organizer>
    </event>
    </root>
```

Complete code

Name: Gayathri E
Roll Number:CB.EN.U4CSE17420

Student Personal Profile Table
a)DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT root (profile-personal*)>
<!ELEMENT profile-personal (student_id, student_dob,
student_year_of_study, student_dept, student_about_myself,
student_phone, student_school, student_address,
student_socialmedia_1, student_socialmedia_2,
students_socialmedia_3)>

<!ELEMENT student_socialmedia (student_socialmedia_1,
student_socialmedia_2, student_socialmedia_3)>

<!ELEMENT student_id (#PCDATA)>
<!ELEMENT student_dob (#PCDATA)>
<!ELEMENT student_year_of_study (#PCDATA)>
<!ELEMENT student_dept (#PCDATA)>
<!ELEMENT student_about_myself (#PCDATA)>
<!ELEMENT student_phone (#PCDATA)>
<!ELEMENT student_school (#PCDATA)>
<!ELEMENT student_address (#PCDATA)>
<!ELEMENT student_socialmedia_1 (#PCDATA)>
<!ELEMENT student_socialmedia_2 (#PCDATA)>
<!ELEMENT student_socialmedia_3 (#PCDATA)>
```

b) XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
    <xs:element name="root" type="rootType" />

    <xs:complexType name="rootType">
        <xs:sequence minOccurs="0" maxOccurs="unbounded">
            <xs:element name="profile-personal"
type="profile-personalType"></xs:element>
        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="profile-personalType">
        <xs:sequence maxOccurs="1">

            <xs:element name="student_id" type="IDType" />
            <xs:element name="student_dob" type="dateType"/>
            <xs:element name="student_year_of_study" type="yearType"
/>
            <xs:element name="student_dept" type="descType" />
            <xs:element name="student_about_myself" type="descType"
/>
            <xs:element name="student_phone" type="phoneType" />
            <xs:element name="student_school" type="descType" />
            <xs:element name="student_address" type="descType" />
                                        <xs:element
name="student_socialmedia" type="mediaType" />

        </xs:sequence>
    </xs:complexType>

    <xs:complexType name="mediaType">
        <xs:sequence>

            <xs:element name="student_socialmedia_1" type="urlType"
/>
            <xs:element name="student_socialmedia_2" type="urlType"
/>
            <xs:element name="student_socialmedia_3" type="urlType"
/>

        </xs:sequence>
    </xs:complexType>

    <xs:simpleType name="phoneType">
        <xs:restriction base="xs:integer">
            <xs:pattern value="[1-9][0-9]{9}"></xs:pattern>
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="urlType">
        <xs:restriction base="xs:anyURI">
            <xs:pattern value='https?://.+'/>
            <!-- accepts only http:// or https:// URIs. -->
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="yearType">
        <xs:restriction base="xs:integer">
            <xs:pattern value="\d{1}"/>
                                        <xs:minInclusive
value="1"></xs:minInclusive>
                                        <xs:maxInclusive
value="5"></xs:maxInclusive>
        </xs:restriction>
    </xs:simpleType>


    <xs:simpleType name="dateType">
        <xs:restriction base="xs:date" />
    </xs:simpleType>


    <xs:simpleType name="IDType">
        <xs:restriction base="xs:integer">
            <xs:minInclusive value="1"></xs:minInclusive>
        </xs:restriction>
    </xs:simpleType>

    <xs:simpleType name="nameType">
        <xs:restriction base="xs:string" />
    </xs:simpleType>

    <xs:simpleType name="descType">
        <xs:restriction base="xs:string">
            <xs:minLength value="10"></xs:minLength>
            <xs:maxLength value="1000"></xs:maxLength>
        </xs:restriction>
    </xs:simpleType>

</xs:schema>
```

C)XSLT

```
<?xml version="1.0" encoding="UTF-8"?>

<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

    <xsl:template match="/">
        <html>
            <head>
                <link rel="stylesheet"
href="https://www.w3schools.com/w3css/4/w3.css" />
                <!--
                <link rel="stylesheet" href="../css/tablecss.css" />
                <link rel="stylesheet" href="../css/style.css" />
                -->
                <link rel="stylesheet" href="tablecss.css" />
                <link rel="stylesheet" href="style.css" />
                <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Sofia" />
            </head>
            <body>
                <center>
                    <h2>Student Personal profile</h2>
                </center>
                <table class="table_props" id="table-title">
                    <tr id="table-heading">

                        <th>Stu_ID</th>
                        <th>DOB</th>
                        <th>Curr_Year</th>
                        <th>Department</th>
                        <th>About</th>
                        <th>Phone Number</th>
```

```
                <th>School</th>
                <th>Student_address</th>
                <th>Socialmedia_1</th>
                <th>Socialmedia_2</th>
                <th>Socialmedia_3</th>

</tr>
    <xsl:for-each select="root/profile-personal">
<tr>
    <td>                    <xsl:value-of select="student_id" />
    </td>
    <td>                    <xsl:value-of select="student_dob" />
    </td>
    <td>                    <xsl:value-of
select="student_year_of_study" />
    </td>
    <td>
<xsl:value-of select="student_dept" />
    </td>
    <td>                    <xsl:value-of select="student_about_myself"
/>
    </td>
    <td>
<xsl:value-of select="student_phone" />
    </td>
    <td>
<xsl:value-of select="student_school" />
    </td>
    <td>
<xsl:value-of select="student_address" />
    </td>
    <td>
                    <xsl:value-of
select="student_socialmedia/student_socialmedia_1" />
    </td>
    <td>
                    <xsl:value-of
select="student_socialmedia/student_socialmedia_2" />
     </td>
     <td>                <xsl:value-of
select="student_socialmedia/student_socialmedia_3" />
     </td>
     </tr>
            </xsl:for-each>
          </table>
        </body>
      </html>
   </xsl:template>
</xsl:stylesheet>
```

C)JDBC Code

```
import java.io.* ;
import java.util.* ;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import org.w3c.dom.Node;
import org.w3c.dom.Element;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;

public class xml_to_mysql_personal{
public static void main ( String [] args ) throws Exception {

                    Class.forName ("com.mysql.cj.jdbc.Driver");
                    Connection con = (Connection)
DriverManager.getConnection("jdbc:mysql://database-1.c4hq5iosxryf.
us-east-1.rds.amazonaws.com/fiesta", "admin", "nithin_aakash");

System.out.println("JDBC CONNECTION SUCCESSFUL!");

Statement stmt = con.createStatement ();

                    DocumentBuilderFactory docBuilderFactory =
DocumentBuilderFactory.newInstance ();
                    DocumentBuilder docBuilder =
docBuilderFactory.newDocumentBuilder ();
                    Document doc = docBuilder.parse (new
File("src/student_profile_personal.xml "));
                    doc.getDocumentElement().normalize();
                    NodeList listOfrecords =
doc.getElementsByTagName("profile-personal");

for ( int s = 0 ; s < listOfrecords.getLength(); s++) {
int id = 0, year = 0;
String temp = "", dob = "", dept="", about="", school="";
String m1 = "", m2 = "", m3 = "", phone = "", address = "";

Node firstrecordNode = listOfrecords.item(s);

Element firstrecordElement = (Element) firstrecordNode ;

try {
NodeList idList =
firstrecordElement.getElementsByTagName("student_id");
Element idElement = (Element) idList.item(0);
NodeList textidList = idElement . getChildNodes ();
temp = ((Node) textidList.item(0)).getNodeValue().trim();
id = Integer.parseInt(temp);
}
catch(Exception e){
System.out.println("Student ID cannot be null");
                    System.exit(0);
}

try {
NodeList dobList = firstrecordElement.getElementsByTagName
("student_dob");
Element dobElement = (Element) dobList.item(0);
NodeList textdobList = dobElement.getChildNodes();
dob = ((Node) textdobList.item(0)).getNodeValue().trim();
}
catch(Exception e) {
System.out.println("Student DOB cannot be null");
        System.exit(0);
}

try {
NodeList yearList = firstrecordElement.getElementsByTagName
("student_year_of_study");
Element yearElement = (Element) yearList.item(0);
NodeList textyearList = yearElement.getChildNodes();
temp = ((Node) textyearList.item(0)).getNodeValue().trim();
year = Integer.parseInt(temp);
}
catch(Exception e) {
System.out.println("Year of study cannot be null");
                    System.exit(0);
}

try {
NodeList deptList = firstrecordElement.getElementsByTagName
("student_dept");
Element deptElement = (Element) deptList.item(0);
NodeList textdeptList = deptElement.getChildNodes();
dept = ((Node) textdeptList.item(0)).getNodeValue().trim();
}
catch(Exception e) {            System.out.println("Student
department cannot be null");            System.exit(0);
}

try {
NodeList aboutList = firstrecordElement.getElementsByTagName
("student_about_myself");
Element aboutElement = (Element) aboutList.item(0);
NodeList textaboutList = aboutElement.getChildNodes();
about = ((Node) textaboutList.item(0)).getNodeValue().trim();
}
catch(Exception e) {            System.out.println("About student
cannot be null");
System.exit(0);
}
try {
NodeList phoneList = firstrecordElement.getElementsByTagName
("student_phone");
Element phoneElement = (Element) phoneList.item(0);
NodeList textphoneList = phoneElement.getChildNodes();
phone = ((Node) textphoneList.item(0)).getNodeValue().trim();
}
catch(Exception e) {
System.out.println("Phone value not provided, is null for the stu_id: "
+ id );
}
```

```
try {
NodeList schoolList = firstrecordElement.getElementsByTagName
("student_school");
Element schoolElement = (Element) schoolList.item(0);
NodeList textschoolList = schoolElement.getChildNodes();
school = ((Node) textschoolList.item(0)).getNodeValue().trim();
}
catch(Exception e) {
System.out.println("School of student cannot be null");
System.exit(0);
}

try {
NodeList addressList = firstrecordElement.getElementsByTagName
("student_address");
Element addressElement = (Element) addressList.item(0);
NodeList textaddressList = addressElement.getChildNodes();
address = ((Node) textaddressList.item(0)).getNodeValue().trim();
}
catch(Exception e) {
System.out.println("Address value not provided, is null for the stu_id:
" + id );
}

try {
NodeList m1List = firstrecordElement.getElementsByTagName
("student_socialmedia_1");
Element m1Element = (Element) m1List.item(0);
NodeList textm1List = m1Element.getChildNodes();
m1 = ((Node) textm1List.item(0)).getNodeValue().trim();
}
catch(Exception e) {                    System.out.println("Social media 1
not provided, is null for the stu_id: " + id);
}

try {
NodeList m2List = firstrecordElement.getElementsByTagName
("student_socialmedia_2");
Element m2Element = (Element) m2List.item(0);
NodeList textm2List = m2Element.getChildNodes();
m2 = ((Node) textm2List.item(0)).getNodeValue().trim();
}
catch(Exception e) {
System.out.println("Social media 2 not provided, is null for the stu_id:
" + id);
}

try {
NodeList m3List = firstrecordElement.getElementsByTagName
("student_socialmedia_3");
Element m3Element = (Element) m3List.item(0);
NodeList textm3List = m3Element.getChildNodes();
m3 = ((Node) textm3List.item(0)).getNodeValue().trim();
}
catch(Exception e) {
System.out.println("Social media 3 not provided, is null for the stu_id:
" + id);
}

String query = "insert into table_student_profile_personal values(' " +
id + " ',' " + dob + " ',' " + year + " ',' " + dept + " ',' " + about + " ',' " +
phone + " ',' " + school + " ',' " + address + " ',' " + m1 + " ',' " + m2 + "
',' " + m3 + "') ";


if ( stmt.executeUpdate(query) == 1 ){
System.out.println(" Successful entry of row " + (s+1));
} else {
System.out.println(" Execution Failed at row " + (s+1));
                }
}
con.close ();
                        System.out.println("SUCCESSFUL
INSERTION FROM XML TO DATABASE ");
                }
}
```

Link for Code Reference:
https://drive.google.com/drive/folders/13CPGMjJzpMxcGFbqgGdBRB4Qgvx6MAYO?usp=sharing

Name: Ilam. Prathyusha

Roll Number:CB.EN.U4CSE17424

3)Student Registration Table

a)DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE student [
        <!ELEMENT student (row)*>
        <!ELEMENT row (id, name, rollnumber, gmail,
password)>
        <!ELEMENT id (#PCDATA)>
        <!ELEMENT name (#PCDATA)>
        <!ELEMENT rollnumber (#PCDATA)>
        <!ELEMENT gmail (#PCDATA)>
        <!ELEMENT password (#PCDATA)>
]>
```

b) XSD

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<xs:schema targetNamespace = "localfile" xmlns = "localfile"
elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="student">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="row" maxOccurs="unbounded"
minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element type="xs:byte" name="id"/>
              <xs:element type="xs:string" name="name"/>
              <xs:element type="xs:byte" name="rollnumber"/>
              <xs:element type="xs:string" name="gmail"/>
              <xs:element type="xs:string" name="password"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

C)XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8"/>
    <meta name="viewport" content="width=device-width,
initial-scale=1.0"/>
    <title>Document</title>
    <style>
      body{
       color: white;
      }
      table, th, td{
           border: 1px solid white;
      }
    </style>
</head>
<body>
    <h1>Students Registered:</h1>
    <table>
        <tr>
          <th>id</th>
          <th>Name</th>
          <th>Rollnumber</th>
          <th>Gmail</th>
          <th>Password</th>
        </tr>
          <xsl:for-each select="student/row">
          <tr>
            <td><xsl:value-of select="id"/></td>
            <td><xsl:value-of select="name"/></td>
            <td><xsl:value-of select="rollnumber"/></td>
            <td><xsl:value-of select="gmail"/></td>
            <td><xsl:value-of select="password"/></td>
          </tr>
        </xsl:for-each>
```

```
   </table>
   <div>
    <h1>List of Students:</h1>
     <xsl:for-each select="student/row">
          <ul>
            <li><xsl:value-of select="name"/></li>
          </ul>
     </xsl:for-each>
   </div>

</body>
</html>

</xsl:template>
</xsl:stylesheet>
```

d) JDBC Code:

```
package project_xml_to_db;

import java.io.File;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.Statement;
import java.util.Arrays;
import java.util.List;

import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.xpath.XPath;
import javax.xml.xpath.XPathConstants;
import javax.xml.xpath.XPathFactory;

import org.w3c.dom.Document;
import org.w3c.dom.Node;
import org.w3c.dom.NodeList;

public class student {
	static private String getTextContent(Node
parentNode,String childName) {
	    NodeList nlist = parentNode.getChildNodes();
	    for (int i = 0 ; i < nlist.getLength() ; i++) {
	    Node n = nlist.item(i);
	    String name = n.getNodeName();
	    if ( name != null && name.equals(childName) )
	       return n.getTextContent();
	    }
	    return "";
	}

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		try {

Class.forName("com.mysql.cj.jdbc.Driver");
			Connection con =
DriverManager.getConnection(

"jdbc:mysql://localhost:3306/project","root","root");
			if (con != null) {

System.out.println("Connected");

			}
			else {
				System.out.println("Not
Connected");
			}




			File file = new
File("src/project_xml_to_db/student.xml");
			DocumentBuilderFactory factory =
DocumentBuilderFactory.newInstance();
			DocumentBuilder builder =
factory.newDocumentBuilder();
			Document xmlDoc =
builder.parse(file);
				NodeList recordslength=
xmlDoc.getElementsByTagName("row");


System.out.println(recordslength.getLength());

				XPath xpath =
XPathFactory.newInstance().newXPath();
				Object res =
xpath.evaluate("/student/row",
					xmlDoc,

XPathConstants.NODESET);

				PreparedStatement stmt = con

.prepareStatement("insert into student values(?,?,?,?,?)");

				for (int i = 0 ; i <
recordslength.getLength() ; i++) {
					Node node =
recordslength.item(i);
//					if(getAttrValue(node,
"name")!=null) {

List<String>columns = Arrays

.asList(getTextContent(node, "id"),

getTextContent(node, "name"),

getTextContent(node, "rollnumber"),

getTextContent(node, "gmail"),

getTextContent(node, "password")

);
						for (int n =
0 ; n < columns.size() ; n++) {

stmt.setString(n+1, columns.get(n));
						}
		stmt.execute();
					}
//				}

			}
			catch (Exception e) {
				System.out.println(e);
			}
		}
}

public class student_skill {
	static private String getTextContent(Node
parentNode,String childName) {
	    NodeList nlist = parentNode.getChildNodes();
	    for (int i = 0 ; i < nlist.getLength() ; i++) {
	    Node n = nlist.item(i);
	    String name = n.getNodeName();
	    if ( name != null && name.equals(childName) )
	       return n.getTextContent();
	    }
	    return "";
	}

	public static void main(String[] args) {
		// TODO Auto-generated method stub
		try {

Class.forName("com.mysql.cj.jdbc.Driver");
			Connection con =
DriverManager.getConnection(

"jdbc:mysql://localhost:3306/project","root","root");
			if (con != null) {

System.out.println("Connected");

			}
			else {
```

```
                                        System.out.println("Not
Connected");
                                }




                                File file = new
File("src/project_xml_to_db/student_skill.xml");
                                DocumentBuilderFactory factory =
DocumentBuilderFactory.newInstance();
                                DocumentBuilder builder =
factory.newDocumentBuilder();
                                Document xmlDoc =
builder.parse(file);


                                NodeList recordslength=
xmlDoc.getElementsByTagName("student");

                                System.out.println("Number of
records available in xml: "+recordslength.getLength());

                                XPath xpath =
XPathFactory.newInstance().newXPath();
                                Object res =
xpath.evaluate("/skillset/student",
                                        xmlDoc,

XPathConstants.NODESET);

                                PreparedStatement stmt = con

.prepareStatement("insert into studentskill values(?,?,?)");

                                for (int i = 0 ; i <
recordslength.getLength() ; i++) {
                                        Node node =
recordslength.item(i);
//                                      if(getAttrValue(node,
"name")!=null) {

List<String>columns = Arrays

.asList(getTextContent(node, "student_id"),

getTextContent(node, "primary_skill"),

getTextContent(node, "ps_rating")

);
                                                for (int n =
0 ; n < columns.size() ; n++) {

stmt.setString(n+1, columns.get(n));
                                                }

stmt.execute();
                                        }
//                              }

                        }
                        catch (Exception e) {
                                System.out.println(e);
                        }

        }

}
public class team_details {
        static private String getTextContent(Node
parentNode,String childName) {
            NodeList nlist = parentNode.getChildNodes();
            for (int i = 0 ; i < nlist.getLength() ; i++) {
            Node n = nlist.item(i);
            String name = n.getNodeName();
            if ( name != null && name.equals(childName) )
                return n.getTextContent();
            }
            return "";
        }
    public static void main(String[] args) {
                // TODO Auto-generated method stub
                try {

Class.forName("com.mysql.cj.jdbc.Driver");
                Connection con =
DriverManager.getConnection(

"jdbc:mysql://localhost:3306/project","root","root");
                if (con != null) {

System.out.println("Connected");

                }
                else {
                        System.out.println("Not
Connected");
                }




                File file = new
File("src/project_xml_to_db/team_details.xml");
                DocumentBuilderFactory factory =
DocumentBuilderFactory.newInstance();
                DocumentBuilder builder =
factory.newDocumentBuilder();
                Document xmlDoc =
builder.parse(file);


                NodeList recordslength=
xmlDoc.getElementsByTagName("row");

                System.out.println("Number of
records available in xml: "+recordslength.getLength());

                XPath xpath =
XPathFactory.newInstance().newXPath();
                Object res =
xpath.evaluate("/team/row",
                        xmlDoc,

XPathConstants.NODESET);

                PreparedStatement stmt = con

.prepareStatement("insert into team_details values(?,?,?)");

                for (int i = 0 ; i <
recordslength.getLength() ; i++) {
                        Node node =
recordslength.item(i);
//                      if(getAttrValue(node,
"name")!=null) {

List<String>columns = Arrays

.asList(getTextContent(node, "id"),

getTextContent(node, "name"),

getTextContent(node, "student_id")

);
                                        for (int n =
0 ; n < columns.size() ; n++) {

stmt.setString(n+1, columns.get(n));
                                        }

stmt.execute();
                                }
//                      }

                }
                catch (Exception e) {
                        System.out.println(e);
                }
```

```
        }
}
```

Link for Code Reference:
https://drive.google.com/drive/folders/12A53sW0MCN9pKEyY3X4BbUQ9cpGQCtol?usp=sharing

Name: Neeraj Kumar Reddy P
Roll No: CB.EN.U4CSE17445

Table_event_registration

a) DTD:

```
<!ELEMENT root (event_id, team_id, registration_datetime)>
<!ELEMENT event_id (#PCDATA)>
<!ELEMENT team_id (#PCDATA)>
<!ELEMENT registration_datetime (#PCDATA)>
```

b) XSD:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
   <xs:element name="root" type="rootType" />
   <xs:complexType name="rootType">
      <xs:sequence>
         <xs:element name="event_id" type="IDType" />
         <xs:element name="team_id" type="IDType" />
         <xs:element name = "registration_datetime" type = "xs:date"/>
      </xs:sequence>
   </xs:complexType>
   <xs:simpleType name="IDType">
      <xs:restriction base="xs:integer">
         <xs:minInclusive value="0"></xs:minInclusive>
         <xs:maxInclusive value="100"></xs:maxInclusive>
      </xs:restriction>
   </xs:simpleType>

   <xs:simpleType name="registration_datetime">
      <xs:restriction base="xs:date">
      </xs:restriction>
   </xs:simpleType>
</xs:schema>
```

c) XSL:

```
<?xml version="1.0" encoding="UTF-8"?
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
   <xsl:template match="/">
      <html>
         <head>
            <link rel="stylesheet"
href="https://www.w3schools.com/w3css/4/w3.css" />
            <link rel="stylesheet" href="tablecss.css" />
            <link rel="stylesheet" href="style.css" />
            <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Sofia" />
         </head>
         <body>
            <center>
               <h2>Table Event Registration</h2>
            </center>
            <table class="table_props" id="table-title">
               <tr id="table-heading">
                  <th>Event_ID</th>
                  <th>Team_ID</th>
                  <th>Registration_datetime</th>
               </tr>
               <xsl:for-each select="root">
                  <tr>
                     <td>
                        <xsl:value-of select="event_id" />
                     </td>
                     <td>
                        <xsl:value-of select="team_id" />
                     </td>
                     <td>
                        <xsl:value-of select="registration_datetime" />
                     </td>
                  </tr>
               </xsl:for-each>
            </table>
         </body>
      </html>
   </xsl:template>
</xsl:stylesheet>
```

d) JDBC Code:

```
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import org.w3c.dom.Node;
import org.w3c.dom.Element;
import java.sql.*;
import java.io.File;

public class table_event_registration {
        public static void main(String[] args) throws Exception {
                Class.forName("com.mysql.cj.jdbc.Driver");
                Connection con =
DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/review2",
"root", "neeraj11");
                if (con != null)
                        System.out.println("Database
connected!");
                else
                        System.out.println("Database not
connected!")
      File id_counter_xmlfile = new
File("src/table_event_Registration.xml"); //Read specific XML file
                DocumentBuilderFactory docBuilderFactory =
DocumentBuilderFactory.newInstance(); //parsing XML data
                DocumentBuilder docBuilder =
docBuilderFactory.newDocumentBuilder();
                Document doc =
docBuilder.parse(id_counter_xmlfile);
        doc.getDocumentElement().normalize();
                NodeList rows =
doc.getElementsByTagName("root"); //Extraction of XML data

                for (int i = 0; i < rows.getLength(); i++) {
                        Node row = rows.item(i);
                        if (row.getNodeType() ==
Node.ELEMENT_NODE) {
                                Element rowElement =
(Element) row;
                                NodeList studentID =
rowElement.getElementsByTagName("event_id");
                                Element
studentElement = (Element) studentID.item(0);
                                NodeList textFNList =
studentElement.getChildNodes();
            String tmp = ((Node)
textFNList.item(0)).getNodeValue().trim();
            int student = Integer.parseInt(tmp);

                                NodeList teamID =
rowElement.getElementsByTagName("team_id");
                                Element teamElement
= (Element) teamID.item(0);
                                NodeList textLNList =
teamElement.getChildNodes();
            tmp = ((Node) textLNList.item(0)).getNodeValue().trim();
            int team =  Integer.parseInt(tmp);
```

```
                                        NodeList eventID =
rowElement.getElementsByTagName("registration_datetime");
                                        Element eventElement
= (Element) eventID.item(0);
                                        NodeList textLNList1 =
eventElement.getChildNodes();
            tmp = ((Node) textLNList1.item(0)).getNodeValue().trim();
            int event = Integer.parseInt(tmp);

                                        String query = "insert
into review2.event_registration values (?,?,?)";
                                        PreparedStatement
stmt = con.prepareStatement(query);
                                        stmt.setInt(1, student);
                                        stmt.setInt(2, team);
                                        stmt.setInt(3, event);
                                        if
(stmt.executeUpdate() == 1) {

System.out.println("XML data is imported and stored in MySQL
Database");
                                         } else {
System.out.println("Unsuccessful");
                                        }
                                    }
                            }
                            con.close();
                }
}
```

Table_Latest_News:

a) DTD:

```
<!ELEMENT root (News_id, News_title, News_message,
News_link)>
<!ELEMENT News_id (#PCDATA)>
<!ELEMENT News_title (#PCDATA)>
<!ELEMENT News_message (#PCDATA)>
<!ELEMENT News_link (#PCDATA)>
```

b) XSD:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
   <xs:element name="root" type="rootType" />
   <xs:complexType name="rootType">
      <xs:sequence>
         <xs:element name = "News_id" type="IDType" />
         <xs:element name = "News_title" type = "xs:string"/>
         <xs:element name = "News_message" type="xs:string" />
         <xs:element name = "News_link" type="urlType" />
      </xs:sequence>
   </xs:complexType>
   <xs:simpleType name="IDType">
      <xs:restriction base="xs:integer">
         <xs:minInclusive value="0"></xs:minInclusive>
      </xs:restriction>
   </xs:simpleType>
   <xs:simpleType name="urlType">
      <xs:restriction base="xs:anyURI">
         <xs:pattern value='https?://.+'/>
      </xs:restriction>
   </xs:simpleType>
</xs:schema>
```

c) XSL:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
   <xsl:template match="/">
      <html>
         <head>
            <link rel="stylesheet"
href="https://www.w3schools.com/w3css/4/w3.css" />
            <link rel="stylesheet" href="tablecss.css" />
            <link rel="stylesheet" href="style.css" />
            <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Sofia" />
         </head>
         <body>
            <center>
               <h2>Table Latest News</h2>
            </center>
            <table class="table_props" id="table-title">
               <tr id="table-heading">
                  <th>News_ID</th>
                  <th>News_Title</th>
                  <th>News_Message</th>
                  <th>News_Link</th>
               </tr>
               <xsl:for-each select="root">
                  <tr>
                     <td>
                        <xsl:value-of select="News_id" />
                     </td>
                     <td>
                        <xsl:value-of select="News_title" />
                     </td>
                     <td>
                        <xsl:value-of select="News_message" />
                     </td>
                     <td>
                     <xsl:value-of select="News_link" />
                  </td>
                  </tr>
               </xsl:for-each>
            </table>
         </body>
      </html>
   </xsl:template>
</xsl:stylesheet>
```

d) JDBC Code:

```
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import org.w3c.dom.Node;
import org.w3c.dom.Element;
import java.sql.*;
import java.io.File;

public class table_latest_news {
            public static void main(String[] args) throws Exception {
                        Class.forName("com.mysql.cj.jdbc.Driver");
                        Connection con =
DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/review2",
"root", "neeraj11");
                        if (con != null)
                                    System.out.println("Database
connected!");
                        else
                                    System.out.println("Database not
connected!");

      File id_counter_xmlfile = new File("src/table_latest_news.xml");

                        DocumentBuilderFactory docBuilderFactory =
DocumentBuilderFactory.newInstance();
                        DocumentBuilder docBuilder =
docBuilderFactory.newDocumentBuilder();
                        Document doc =
docBuilder.parse(id_counter_xmlfile);
                        doc.getDocumentElement().normalize();
```

```
                        NodeList rows =
doc.getElementsByTagName("root");

                        for (int i = 0; i < rows.getLength(); i++) {
                                Node row = rows.item(i);

                                if (row.getNodeType() ==
Node.ELEMENT_NODE) {
                                        Element rowElement =
(Element) row;

                                        NodeList studentID =
rowElement.getElementsByTagName("News_id");
                                        Element
studentElement = (Element) studentID.item(0);
                                        NodeList textFNList =
studentElement.getChildNodes();
            String tmp = ((Node)
textFNList.item(0)).getNodeValue().trim();
            int student = Integer.parseInt(tmp);

                                         NodeList teamID =
rowElement.getElementsByTagName("News_title");
                                        Element teamElement
= (Element) teamID.item(0);
                                        NodeList textLNList =
teamElement.getChildNodes();
            tmp = ((Node) textLNList.item(0)).getNodeValue().trim();
            int team =  Integer.parseInt(tmp);

                                        NodeList eventID =
rowElement.getElementsByTagName("News_message");
                                        Element eventElement
= (Element) eventID.item(0);
                                        NodeList textLNList1 =
eventElement.getChildNodes();
            tmp = ((Node) textLNList1.item(0)).getNodeValue().trim();
            int event = Integer.parseInt(tmp);
            NodeList announcmentID =
rowElement.getElementsByTagName("News_link");
 Element announcementElement = (Element)
announcmentID.item(0);
NodeList textLNList2 = announcementElement.getChildNodes();
             tmp = ((Node) textLNList2.item(0)).getNodeValue().trim();
             int announcement = Integer.parseInt(tmp);
                                        String query = "insert
into review2.latest_news values (?,?,?,?)"; PreparedStatement stmt =
con.prepareStatement(query);
stmt.setInt(1, student);                stmt.setInt(2, team);
stmt.setInt(3, event);
stmt.setInt(4, announcement);
if (stmt.executeUpdate() == 1) {
System.out.println("Successful execution");
                                         } else {
             System.out.println("Unsuccessful");
                                        }}
                        }
                        con.close();
            }
}
```

Table_Student_Scores

a) DTD:

```
<!ELEMENT root (student_id, student_score)>
<!ELEMENT student_id (#PCDATA)>
<!ELEMENT student_score (#PCDATA)>
```

b) XSD:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
   <xs:element name="root" type="rootType" />
   <xs:complexType name="rootType">
      <xs:sequence>
         <xs:element name="student_id" type="IDType" />
         <xs:element name="student_score" type="IDType" />
      </xs:sequence>
   </xs:complexType>
   <xs:simpleType name="IDType">
      <xs:restriction base="xs:integer">
         <xs:minInclusive value="0"></xs:minInclusive>
         <xs:maxInclusive value="100"></xs:maxInclusive>
      </xs:restriction>
   </xs:simpleType>
</xs:schema>
```

c) XSL:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
   <xsl:template match="/">
      <html>
         <head>
            <link rel="stylesheet"
href="https://www.w3schools.com/w3css/4/w3.css" />
            <link rel="stylesheet" href="tablecss.css" />
            <link rel="stylesheet" href="style.css" />
            <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Sofia" />
         </head>
         <body>
            <center>
               <h2>Table Student Scores</h2>
            </center>
            <table class="table_props" id="table-title">
               <tr id="table-heading">
                  <th>Student_ID</th>
                  <th>Student_score</th>
               </tr>
               <xsl:for-each select="root">
                  <tr>
                     <td>
                    <xsl:value-of select="student_id" />
                     </td>
                     <td>
                   <xsl:value-of select="student_score" />
                     </td>
                  </tr>
               </xsl:for-each>
            </table>
         </body>
      </html>
   </xsl:template>
</xsl:stylesheet>
```

d) JDBC Code:

```
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import org.w3c.dom.Node;
import org.w3c.dom.Element;
import java.sql.*;
import java.io.File;
public class table_student_scores {
            public static void main(String[] args) throws Exception {
Class.forName("com.mysql.cj.jdbc.Driver");
Connection con =
DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/review2",
"root", "neeraj11");
                        if (con != null)
System.out.println("Database connected!");
                        else        System.out.println("Database not
connected!");
```

```
        File id_counter_xmlfile = new
File("src/table_student_scores.xml");
 DocumentBuilderFactory docBuilderFactory =
DocumentBuilderFactory.newInstance();
DocumentBuilder docBuilder =
docBuilderFactory.newDocumentBuilder();
Document doc = docBuilder.parse(id_counter_xmlfile);
doc.getDocumentElement().normalize();
NodeList rows = doc.getElementsByTagName("root");
 for (int i = 0; i < rows.getLength(); i++) {
                                Node row = rows.item(i);
if (row.getNodeType() == Node.ELEMENT_NODE) {
Element rowElement = (Element) row;
NodeList studentID =
rowElement.getElementsByTagName("student_id");
Element studentElement = (Element) studentID.item(0);
NodeList textFNList = studentElement.getChildNodes();
                String tmp = ((Node)
textFNList.item(0)).getNodeValue().trim();
          int student = Integer.parseInt(tmp);
NodeList teamID =
rowElement.getElementsByTagName("student_score");
 Element teamElement = (Element) teamID.item(0);
NodeList textLNList = teamElement.getChildNodes();
                tmp = ((Node) textLNList.item(0)).getNodeValue().trim();
                int team =  Integer.parseInt(tmp);
String query = "insert into review2.student_scores values (?,?)";
PreparedStatement stmt = con.prepareStatement(query);
stmt.setInt(1, student);
stmt.setInt(2, team);
if (stmt.executeUpdate() == 1) {     System.out.println("Successful
execution");
                                                        } else {

System.out.println("Unsuccessful");
                                                        }
                                        }
                                }
                                con.close();
                }
}
```

Drive Link for Code:
https://drive.google.com/file/d/1YT-LTZpp2zuKx7mDKOYPPvdguW8nWoP/view?usp=sharing

Name : Sanjay Tharagesh R. S.
Roll : CB.EN.U4CSE17453

9) Contact Us Table

a) DTD

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE contactus [
                <!ELEMENT contactus (row)*>
                <!ELEMENT row (contact_id, contact_name,
contact_phone, contact_email, contact_message)>
                <!ELEMENT contact_id (#PCDATA)>
                <!ELEMENT contact_name (#PCDATA)>
                <!ELEMENT contact_phone (#PCDATA)>
                <!ELEMENT contact_email (#PCDATA)>
                <!ELEMENT contact_message (#PCDATA)>
]>
```

b) XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
   <xs:element name="contactus">
      <xs:complexType>
         <xs:sequence>
            <xs:element maxOccurs="unbounded" ref="row"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:element name="row">
      <xs:complexType>
         <xs:sequence>
            <xs:element ref="contact_id"/>
            <xs:element ref="contact_name"/>
            <xs:element ref="contact_phone"/>
            <xs:element ref="contact_email"/>
            <xs:element ref="contact_message"/>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
               <xs:element name="contact_id">
      <xs:simpleType>
         <xs:restriction base="xs:integer">
            <xs:minInclusive value="1"/>
         </xs:restriction>
      </xs:simpleType>
   </xs:element>
   <xs:element name="contact_name" type="xs:string"/>
               <xs:element name="contact_phone">
      <xs:simpleType>
         <xs:restriction base="xs:string">
            <xs:pattern value="[1-9][0-9]{9}"/>
         </xs:restriction>
      </xs:simpleType>
   </xs:element>
               <xs:element name="contact_email">
      <xs:simpleType>
         <xs:restriction base="xs:string">
            <xs:pattern
value="([0-9a-zA-Z]([-.\w]*[0-9a-zA-Z])*@([0-9a-zA-Z][-\w]*[0-9a-zA-Z
]\.)+[a-zA-Z]{2,9})"/>
         </xs:restriction>
      </xs:simpleType>
   </xs:element>
               <xs:element name="contact_message">
      <xs:simpleType>
         <xs:restriction base="xs:string">
            <xs:whiteSpace value="preserve"/>
         </xs:restriction>
      </xs:simpleType>
   </xs:element>

</xs:schema>
```

c) XSL

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
   xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<head>
   <style>
      table {
      width: 900px;
      border-collapse: collapse;
      margin:50px auto;
      }
      tr:nth-of-type(odd) {
      background: #eee;
      }
      th {
      background: #3498db;
      color: white;
      font-weight: bold;
      }
      td, th {
      padding: 10px;
      border: 1px solid #ccc;
      text-align: left;
      font-size: 15px;
      font-family: "Verdana";
      }
   </style>
</head>
<body style="text-align:center">
   <table>
      <tr>
         <th>Contact ID</th>
         <th>Name</th>
         <th>Phone</th>
         <th>Email</th>
         <th>Message</th>
      </tr>
```

```
    <xsl:for-each select="contactus/row">
      <tr>
        <td>
          <xsl:value-of select="contact_id"/>
        </td>
        <td>
          <xsl:value-of select="contact_name"/>
        </td>
        <td>
          <xsl:value-of select="contact_phone"/>
        </td>
        <td>
          <xsl:value-of select="contact_email"/>
        </td>
        <td>
          <xsl:value-of select="contact_message"/>
        </td>
      </tr>
    </xsl:for-each>
  </table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

10) Announcement Table

a) DTD

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE announcements [
        <!ELEMENT announcements (row)*>
        <!ELEMENT row (announcement_id, announcement_title,
announcement_message, event_id)>
        <!ELEMENT announcement_id (#PCDATA)>
        <!ELEMENT announcement_title (#PCDATA)>
        <!ELEMENT announcement_message (#PCDATA)>
        <!ELEMENT event_id (#PCDATA)>
]>
```

b) XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <xs:element name="announcements">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="row"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="row">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="announcement_id"/>
        <xs:element ref="announcement_title"/>
        <xs:element ref="announcement_message"/>
        <xs:element ref="event_id"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
        <xs:element name="announcement_id">
    <xs:simpleType>
      <xs:restriction base="xs:integer">
        <xs:minInclusive value="1"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="announcement_title" type="xs:string"/>
        <xs:element name="announcement_message">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:whiteSpace value="preserve"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
        <xs:element name="event_id">
    <xs:simpleType>
      <xs:restriction base="xs:integer">
        <xs:minInclusive value="1"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
</xs:schema>
```

c) XSL

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<head>
        <style>
                table {
                width: 900px;
                border-collapse: collapse;
                margin:50px auto;
                }
                tr:nth-of-type(odd) {
                background: #eee;
                }
                th {
                background: #3498db;
                color: white;
                font-weight: bold;
                }
                td, th {
                padding: 10px;
                border: 1px solid #ccc;
                text-align: left;
                font-size: 15px;
                font-family: "Verdana";
                }
        </style>
</head>
<body style="text-align:center">
        <table>
                <tr>
                        <th>Announcement ID</th>
                        <th>Title</th>
                        <th>Message</th>
                        <th>Event ID</th>
                </tr>
                <xsl:for-each select="announcements/row">
                        <tr>
                                <td>

<xsl:value-of select="announcement_id"/>
                                </td>
                                <td>

<xsl:value-of select="announcement_title"/>
                                </td>
                                <td>

<xsl:value-of select="announcement_message"/>
                                </td>
                                <td>

<xsl:value-of select="event_id"/>
                                </td>
                        </tr>
                </xsl:for-each>
        </table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

11) Event Winners Table

a) DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE event_winners [
        <!ELEMENT event_winners (row)*>
        <!ELEMENT row (event_id, team_id, position)>
        <!ELEMENT event_id (#PCDATA)>
        <!ELEMENT team_id (#PCDATA)>
        <!ELEMENT position (#PCDATA)>
]>
```

b) XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <xs:element name="event_winners">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="row"/>
      </xs:sequence>
```

```
    </xs:complexType>
  </xs:element>
  <xs:element name="row">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="event_id"/>
        <xs:element ref="team_id"/>
        <xs:element ref="position"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
          <xs:element name="event_id">
    <xs:simpleType>
      <xs:restriction base="xs:integer">
        <xs:minInclusive value="1"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
          <xs:element name="team_id">
    <xs:simpleType>
      <xs:restriction base="xs:integer">
        <xs:minInclusive value="1"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
          <xs:element name="position">
    <xs:simpleType>
      <xs:restriction base="xs:integer">
        <xs:minInclusive value="1"/>
                                        <xs:maxInclusive
value="10"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
</xs:schema>
```

c) XSL

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<head>
        <style>
                table {
                width: 900px;
                border-collapse: collapse;
                margin:50px auto;
                }
                tr:nth-of-type(odd) {
                background: #eee;
                }
                th {
                background: #3498db;
                color: white;
                font-weight: bold;
                }
                td, th {
                padding: 10px;
                border: 1px solid #ccc;
                text-align: left;
                font-size: 15px;
                font-family: "Verdana";
                }
        </style>
</head>
<body style="text-align:center">
        <table>
                <tr>
                        <th>Event ID</th>
                        <th>Team ID</th>
                        <th>Position</th>
                </tr>
                <xsl:for-each select="event_winners/row">
                        <tr>
                                <td>

<xsl:value-of select="event_id"/>
                                </td>
                                <td>

<xsl:value-of select="team_id"/>
                                </td>
                                <td>

<xsl:value-of select="position"/>
                                </td>
                        </tr>
                </xsl:for-each>
        </table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

All JDBC Code (1 code for all tables)

```
package ncp_project;

import java.sql.*;
import java.io.*;
import java.util.*;
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.xpath.XPath;
import javax.xml.xpath.XPathConstants;
import javax.xml.xpath.XPathFactory;
import javax.xml.parsers.DocumentBuilder;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import org.w3c.dom.Node;
import org.w3c.dom.Element;
import org.w3c.dom.NamedNodeMap;

public class PushToSQL {

        static private String getAttrValue(Node node, String
attrName) {
                if(!node.hasAttributes())
                        return "";
                NamedNodeMap nmap = node.getAttributes();
                if(nmap == null)
                        return "";
                Node n = nmap.getNamedItem(attrName);
                if(n == null)
                        return "";
                return n.getNodeValue();
        }

        static private String getTextContent(Node parentNode,
String childName) {
                NodeList nlist = parentNode.getChildNodes();
                for(int i=0; i<nlist.getLength(); i++) {
                        Node n = nlist.item(i);
                        String name = n.getNodeName();
                        if(name != null &&
name.equals(childName))
                                return
n.getTextContent();
                }
                return "";
        }

        public static void main(String[] args) {
                // TODO Auto-generated method stub
                try {

Class.forName("com.mysql.cj.jdbc.Driver");
                        String path_to_db =
"jdbc:mysql://localhost:3306/fiesta";
                        String username = "root";
                        String password = "root";
                        String xml_file1 =
"D:\\sem7\\NCP\\Project\\clones\\fiesta\\Review2_XML\\Sanjay\\4 -
Table_Contactus\\xml_dtd\\4_table_contactus.xml";
                        String xml_file2 =
"D:\\sem7\\NCP\\Project\\clones\\fiesta\\Review2_XML\\Sanjay\\5 -
Table_Announcement\\xml_dtd\\5_table_announcement.xml";
                        String xml_file3 =
"D:\\sem7\\NCP\\Project\\clones\\fiesta\\Review2_XML\\Sanjay\\9 -
Table_Event_Winners\\xml_dtd\\9_table_event_winners.xml";
                        Connection con =
DriverManager.getConnection(path_to_db, username, password);

                        File file1 = new File(xml_file1);
                        File file2 = new File(xml_file2);
                        File file3 = new File(xml_file3);
```

```
                                DocumentBuilderFactory factory =
DocumentBuilderFactory.newInstance();
                                DocumentBuilder builder =
factory.newDocumentBuilder();

                                Document xmlDoc1 =
builder.parse(file1);
                                Document xmlDoc2 =
builder.parse(file2);
                                Document xmlDoc3 =
builder.parse(file3);

                                NodeList nlist1 =
xmlDoc1.getElementsByTagName("row");
                                NodeList nlist2 =
xmlDoc2.getElementsByTagName("row");
                                NodeList nlist3 =
xmlDoc3.getElementsByTagName("row");

                                XPath xpath =
XPathFactory.newInstance().newXPath();
                                Object res1 =
xpath.evaluate("/contactus/row", xmlDoc1,
XPathConstants.NODESET);
                                Object res2 =
xpath.evaluate("/announcements/row", xmlDoc2,
XPathConstants.NODESET);
                                Object res3 =
xpath.evaluate("/event_winners/row", xmlDoc3,
XPathConstants.NODESET);

                                PreparedStatement stmt1 =
con.prepareStatement("INSERT INTO table_contactus("
                                                +
"contact_id, contact_name, contact_phone, contact_email,
contact_message)"
                                                +
"VALUES(?, ?, ?, ?, ?)");

                                PreparedStatement stmt2 =
con.prepareStatement("INSERT INTO table_announcement("
                                                +
"announcement_id, announcement_message, announcement_title,
event_id)"
                                                +
"VALUES(?, ?, ?, ?)");

                                PreparedStatement stmt3 =
con.prepareStatement("INSERT INTO table_event_winners("
                                                +
"event_id, position, team_id)"
                                                +
"VALUES(?, ?, ?)");

                                for(int i=0; i<nlist1.getLength();
i++) {
                                        Node node =
nlist1.item(i);
                                        List<String> columns =
Arrays.asList(

getTextContent(node, "contact_id"),

getTextContent(node, "contact_name"),

getTextContent(node, "contact_phone"),

getTextContent(node, "contact_email"),

getTextContent(node, "contact_message")

);
                                        for(int n=0;
n<columns.size(); n++) {

if(columns.get(n) == "")

stmt1.setNull(n+1, java.sql.Types.VARCHAR);
                                                else

stmt1.setString(n+1, columns.get(n));
                                        }
                                        stmt1.execute();
System.out.println("Inserting row " + i);
                                        }

                                        for(int i=0; i<nlist2.getLength();
i++) {
                                                Node node =
nlist2.item(i);
                                                List<String> columns =
Arrays.asList(

getTextContent(node, "announcement_id"),

getTextContent(node, "announcement_message"),

getTextContent(node, "announcement_title"),

getTextContent(node, "event_id")

);
                                for(int n=0; n<columns.size(); n++) {

if(columns.get(n) == "")

stmt2.setNull(n+1, java.sql.Types.INTEGER);
                                        else

stmt2.setString(n+1, columns.get(n));
                                                }
                                                stmt2.execute();

System.out.println("Inserting row " + i);
                                        }
                                        for(int i=0; i<nlist3.getLength();
i++) {
                                        Node node = nlist3.item(i);
                                List<String> columns = Arrays.asList(

getTextContent(node, "event_id"),

getTextContent(node, "position"),

getTextContent(node, "team_id")

);
for(int n=0; n<columns.size(); n++) {

stmt3.setString(n+1, columns.get(n));
                                                }
stmt3.execute();
System.out.println("Inserting row " + i);
                }
                con.close();
                }
                catch(Exception e) {
                System.out.println(e);
                } }
}
```

Name : Srishilesh P. S.
Roll : CB.EN.U4CSE17458

14) table_event

a) DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT root (event*)>
<!ELEMENT event (event_id, event_name, event_desc, event_tags,
event_type_participation, event_timing, event_venue,
event_max_participants, event_registration_link,event_organizer)>
<!ELEMENT event_tags (tag*)>
<!ELEMENT tag (#PCDATA)>
<!ATTLIST tag id CDATA #REQUIRED>
<!ATTLIST tag displayName CDATA #REQUIRED>
<!ELEMENT event_timing (event_start_datetime,
event_end_datetime)>
<!ELEMENT event_start_datetime (#PCDATA)>
<!ELEMENT event_end_datetime (#PCDATA)>
<!ELEMENT event_organizer (event_organizer_intracollege,
event_organizer_other, event_organizer_name,
event_organizer_email, event_organizer_phone,
event_organizer_details, event_organizer_poster)>
<!ELEMENT event_organizer_intracollege (event_organizer_college,
event_organizer_campus)>
```

```
<!ELEMENT event_organizer_college (#PCDATA)>
<!ELEMENT event_organizer_campus (#PCDATA)>
<!ELEMENT event_id (#PCDATA)>
<!ELEMENT event_name (#PCDATA)>
<!ELEMENT event_desc (#PCDATA)>
<!ELEMENT event_type_participation (#PCDATA)>
<!ELEMENT event_venue (#PCDATA)>
<!ELEMENT event_max_participants (#PCDATA)>
<!ELEMENT event_registration_link (#PCDATA)>
<!ELEMENT event_organizer_other (#PCDATA)>
<!ELEMENT event_organizer_name (#PCDATA)>
<!ELEMENT event_organizer_email (#PCDATA)>
<!ELEMENT event_organizer_phone (#PCDATA)>
<!ELEMENT event_organizer_details (#PCDATA)>
<!ELEMENT event_organizer_poster (#PCDATA)>
```

b) XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
    <xs:element name="root" type="rootType" />

    <xs:complexType name="rootType">
<xs:sequence minOccurs="0" maxOccurs="unbounded">
<xs:element name="event" type="eventType"></xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="eventType">
<xs:sequence maxOccurs="1">
<xs:element name="event_id" type="IDType" />
<xs:element name="event_name" type="nameType" />
<xs:element name="event_desc" type="descType" />
<xs:element name="event_tags" type="tagsType" />
<xs:element name="event_type_participation"
type="participationType" />
<xs:element name="event_timing" type="timingType" />
<xs:element name="event_venue" type="nameType" />
<xs:element name="event_max_participants" type="numType" />
<xs:element name="event_registration_link" type="nameType" />
<xs:element name="event_organizer" type="organizerType" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="tagsType">
<xs:sequence maxOccurs="5">
<xs:element name="tag" />
</xs:sequence>
</xs:complexType>
<xs:attribute name="id" >
<xs:simpleType >
<xs:restriction base="xs:string">
</xs:restriction>
</xs:simpleType>
</xs:attribute>
<xs:complexType name="timingType">
<xs:sequence maxOccurs="1" minOccurs="1">
<xs:element name="event_start_datetime"
type="timeType"></xs:element>
<xs:element name="event_end_datetime"
type="timeType"></xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="organizerType">
<xs:sequence>
<xs:element name="event_organizer_intracollege"
type="collegeType"></xs:element>
<xs:element name="event_organizer_other"
type="nameType"></xs:element>
<xs:element name="event_organizer_name"
type="nameType"></xs:element>
<xs:element name="event_organizer_email"
type="emailType"></xs:element>
<xs:element name="event_organizer_phone"
type="phoneType"></xs:element>
<xs:element name="event_organizer_details"
type="nameType"></xs:element>
<xs:element name="event_organizer_poster"></xs:element>
</xs:sequence>
</xs:complexType>
<xs:complexType name="collegeType">
<xs:sequence maxOccurs="1">
<xs:element name="event_organizer_college"
type="nameType"></xs:element>
<xs:element name="event_organizer_campus"
type="nameType"></xs:element>
</xs:sequence>
</xs:complexType>
<xs:simpleType name="emailType">
<xs:restriction base="xs:string">
<xs:pattern
value="\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*"></xs:pattern>
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="phoneType">
<xs:restriction base="xs:string">
<xs:pattern value="\+?\d[\d -]{8,12}\d"></xs:pattern>
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="timeType">
<xs:restriction base="xs:dateTime" />
</xs:simpleType>
<xs:simpleType name="IDType">
<xs:restriction base="xs:integer">
<xs:minInclusive value="0"></xs:minInclusive>
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="nameType">
<xs:restriction base="xs:string" />
</xs:simpleType>
<xs:simpleType name="descType">
<xs:restriction base="xs:string">
<xs:minLength value="10"></xs:minLength>
<xs:maxLength value="1000"></xs:maxLength>
</xs:restriction>
</xs:simpleType>
<xs:simpleType name="participationType">
<xs:restriction base="xs:string">
          <xs:enumeration value="Solo"></xs:enumeration>
          <xs:enumeration value="Team"></xs:enumeration>
      </xs:restriction>
   </xs:simpleType>

   <xs:simpleType name="numType">
      <xs:restriction base="xs:integer">
         <xs:minInclusive value="10"></xs:minInclusive>
         <xs:maxInclusive value="1000"></xs:maxInclusive>
      </xs:restriction>
   </xs:simpleType>

</xs:schema>
```

c) XSL

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:template match="/">
 <html>
 <head>
 <link rel="stylesheet"
href="https://www.w3schools.com/w3css/4/w3.css" />
 <link rel="stylesheet" href="../css/tablecss.css" />
 <!-- <link rel="stylesheet" href="../css/style.css" /> -->
 <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Sofia" />
 </head>
 <body>
 <center>
 <h2>Table Events</h2>
 <p>
 Number of Events - <xsl:value-of
select="count(root/event/event_max_participants)" /><br/>
 </p>
 </center>
 <div class="scrolling">
<table class="table-props" id="table-title">
<tr id="table-heading">
<th>Event_ID</th>
<th>Event Name</th>
<th>Event Description</th>
<th>Event Tags</th>
<th>Participation type</th>
<th>Start date and time</th>
<th>End date and time</th>
<th>Venue</th>
<th>Max no. of participants</th>
<th>Registration link</th>
```

```
<th>Organizer college</th>
<th>Organizer college - campus</th>
<th>Organizer name</th>
<th>Organizer email</th>
<th>Organizer phone</th>
<th>Organizer details</th>
<th>Poster</th>
</tr>
<xsl:for-each select="root/event">
<xsl:sort select="number(event_id)" data-type="number" />
<tr>
<td>
<xsl:value-of select="event_id" />
</td>
<td>
<xsl:value-of select="event_name" />
</td>
<td>
<xsl:value-of select="event_desc" />
</td>
<td>
<xsl:for-each select="event_tags">
      <xsl:call-template name="getTagField">
                              <xsl:with-param name="tag"
select="current()"></xsl:with-param>
                           </xsl:call-template>
                        </xsl:for-each>
                     </td>
                     <xsl:choose>
                        <xsl:when test="event_type_participation =
'Team'">
                           <td bgcolor="#f2f2f242">
                              <xsl:value-of
select="event_type_participation" />
                           </td>
</xsl:when>                        <xsl:otherwise>
<td bgcolor="#cccccc">                              <xsl:value-of
select="event_type_participation" />                  </td>
</xsl:otherwise>
</xsl:choose>
<td>
<xsl:value-of select="event_timing/event_start_datetime" />
</td>
<td>
<xsl:value-of select="event_timing/event_end_datetime" />
</td>
<td>
<xsl:value-of select="event_venue" />
</td>
<td>
<xsl:value-of select="event_max_participants" />
</td>
<td>
<xsl:value-of select="event_registration_link" />
</td>
<td>
<xsl:value-of
select="event_organizer/event_organizer_intracollege/event_organiz
er_college" />
</td>
<td>
<xsl:value-of
select="event_organizer/event_organizer_intracollege/event_organiz
er_campus" />
</td>
<td>
                        <xsl:value-of
select="event_organizer/event_organizer_name" />
                     </td>
                     <td>
                        <xsl:value-of
select="event_organizer/event_organizer_email" />
                     </td>
                     <td>
                        <xsl:value-of
select="event_organizer/event_organizer_phone" />
                     </td>
                     <td>
                        <xsl:value-of
select="event_organizer/event_organizer_details" />
                     </td>
                     <td>
                           <xsl:value-of
select="event_organizer/event_organizer_poster" />
                        </td>
                     </tr>
                  </xsl:for-each>
               </table>
            </div>

         </body>
      </html>
   </xsl:template>
   <xsl:template name="getTagField">
      <xsl:param name="tag" />
      <xsl:variable name="tagIds">
         <xsl:for-each select="//tag">
            <xsl:value-of select="@displayName"></xsl:value-of>
         </xsl:for-each>
      </xsl:variable>
      <xsl:element name="tags">
         <xsl:value-of select="normalize-space(concat($tag,','))" />
      </xsl:element>
   </xsl:template>
</xsl:stylesheet>
```

d) JDBC code

```
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import org.w3c.dom.Node;
import org.w3c.dom.Element;
import java.sql.*;
import java.io.File;
import java.io.FileInputStream;

public class table_event_xml_database_entry {
            public static void main(String[] args) throws Exception {
                        Class.forName("com.mysql.cj.jdbc.Driver");
                        Connection con =
DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/sys",
"root", "root");
                        if (con != null)
                                    System.out.println("Database
connected! \n");
                        else
                                    System.out.println("Database not
connected! \n");

      File event_xmlfile = new File("./src/table_event.xml");

                        DocumentBuilderFactory docBuilderFactory =
DocumentBuilderFactory.newInstance();
                        DocumentBuilder docBuilder =
docBuilderFactory.newDocumentBuilder();
                        Document doc =
docBuilder.parse(event_xmlfile);
                        doc.getDocumentElement().normalize();
                        NodeList events =
doc.getElementsByTagName("event");


                        for (int i = 0; i < events.getLength(); i++) {
                                    int eventid = 0, eventType = 0,
maxParticipants = 10;
                                    String eventname = "", eventdesc
= "", startDateTime = "", endDateTime = "", eventVenue = "",
registrationLink = "";
                                    String organizerCollege = "",
organizerCampus = "", organizerOther = "", organizerName = "";
                                    String organizerEmail = "",
organizerPhone = "", organizerDetails = "", tags = "", poster="";
                                    FileInputStream fileread = null;

                                    Node row = events.item(i);

                                    if (row.getNodeType() ==
Node.ELEMENT_NODE) {
                                                Element rowElement =
(Element) row;

                                                try {
                                                            NodeList
eventID = rowElement.getElementsByTagName("event_id");
```

```
                                                    Element
eventElement = (Element) eventID.item(0);
                                                    NodeList
textFNList = eventElement.getChildNodes();
                                                    String
eventno = ((Node) textFNList.item(0)).getNodeValue().trim();
                                                    eventid =
Integer.parseInt(eventno);
                                            }
                                            catch(Exception e) {

System.out.println("Event ID: null");
                                            }

                                            try {
                                                    NodeList
eventNameList =
rowElement.getElementsByTagName("event_name");
                                                     Element
eventNameElement = (Element) eventNameList.item(0);
                                                     NodeList
textLNList = eventNameElement.getChildNodes();

eventname = ((Node) textLNList.item(0)).getNodeValue().trim();
                                            }
                                            catch (Exception e) {

System.out.println("Event Name: null");
                                            }


                                            try {
                                                    NodeList
eventDescList = rowElement.getElementsByTagName("event_desc");
                                                    Element
eventDescElement = (Element) eventDescList.item(0);
                                                    NodeList
textLNList1 = eventDescElement.getChildNodes();
                                                    eventdesc
= ((Node) textLNList1.item(0)).getNodeValue().trim();
                                            }
                                            catch(Exception e) {

System.out.println("Event Description: null");
                                            }

                                            // WRITE FOR
EVENT_TAGS
                                            try {
                                                    NodeList
eventTagsList = rowElement.getElementsByTagName("event_tags");
                                                    for (int j =
0; j < eventTagsList.getLength(); j ++) {

Node tag = eventTagsList.item(j);


if (tag.getNodeType() == Node.ELEMENT_NODE) {

        Element tagElement = (Element) tag;



        NodeList tagList =
tagElement.getElementsByTagName("tag");

        Element aa = (Element) tagList.item(j);

        NodeList bb = aa.getChildNodes();

        tags += ((Node) bb.item(j)).getNodeValue().trim();

}
                                                    }
//
System.out.println("TAGS LENGTH: " + tags);
                                            }
                                            catch(Exception e) {

System.out.println("Tags: null");
                                            }

                                            try {
                                                    NodeList
eventTypeList =
rowElement.getElementsByTagName("event_type_participation");
                                                    Element
eventTypeElement = (Element) eventTypeList.item(0);
                                                    NodeList
textLNList2 = eventTypeElement.getChildNodes();
                                                    String
eventype = ((Node) textLNList2.item(0)).getNodeValue().trim();
                                                    eventType
= 0;
                                                    if
(eventype.equals("Team"))

eventType = 1;
                                                    else

eventType = 0;
                                            }
                                            catch(Exception e) {

System.out.println("Event Type: null");
                                            }

                                            try {
                                                    NodeList
startdatetimeList =
rowElement.getElementsByTagName("event_start_datetime");
                                                    Element
startdatetimeElement = (Element) startdatetimeList.item(0);
                                                    NodeList
aa = startdatetimeElement.getChildNodes();

startDateTime = ((Node) aa.item(0)).getNodeValue().trim();

                                                    NodeList
enddatetimeList =
rowElement.getElementsByTagName("event_end_datetime");
                                                    Element
enddatetimeElement = (Element) enddatetimeList.item(0);
                                                    NodeList
bb = enddatetimeElement.getChildNodes();

endDateTime = ((Node) bb.item(0)).getNodeValue().trim();


                                            }
                                            catch(Exception e) {

System.out.println("Date & Time: null");
                                            }

                                            try {
                                                    NodeList
eventVenueList =
rowElement.getElementsByTagName("event_venue");
                                                    Element
eventVenueElement = (Element) eventVenueList.item(0);
                                                    NodeList
cc = eventVenueElement.getChildNodes();

eventVenue = ((Node) cc.item(0)).getNodeValue().trim();
                                            }
                                            catch(Exception e) {

System.out.println("Event Venue: null");
                                            }

                                            try {
                                                    NodeList
eventMaxList =
rowElement.getElementsByTagName("event_max_participants");
                                                    Element
eventMaxElement = (Element) eventMaxList.item(0);
                                                    NodeList
textLNList4 = eventMaxElement.getChildNodes();
                                                    String
maxparticipants = ((Node) textLNList4.item(0)).getNodeValue().trim();

maxParticipants = Integer.parseInt(maxparticipants);
                                            }
                                            catch(Exception e) {

System.out.println("Max participants: null");
```

```
                                        }

                                        try {
                                                NodeList
linkList =
rowElement.getElementsByTagName("event_registration_link");
                                                Element
linkElement = (Element) linkList.item(0);
                                                NodeList
textLNList5 = linkElement.getChildNodes();

registrationLink = ((Node) textLNList5.item(0)).getNodeValue().trim();
                                        }
                                        catch(Exception e) {

System.out.println("Registration link: null");
                                        }

                                        try {
                                                NodeList
organizerCollegeList =
rowElement.getElementsByTagName("event_organizer_college");
                                                Element
organizerCollegeElement = (Element) organizerCollegeList.item(0);
                                                NodeList
dd = organizerCollegeElement.getChildNodes();

organizerCollege = ((Node) dd.item(0)).getNodeValue().trim();
                                        }
                                        catch(Exception e) {

System.out.println("Organizing college: null");
                                        }

                                        try {
                                                NodeList
organizerCampusList =
rowElement.getElementsByTagName("event_organizer_campus");
                                                Element
organizerCampusElement = (Element) organizerCampusList.item(0);
                                                NodeList
ee = organizerCampusElement.getChildNodes();

organizerCampus = ((Node) ee.item(0)).getNodeValue().trim();
                                        }
                                        catch(Exception e) {

System.out.println("Organizing campus: null");
                                        }


                                        try {
                                                NodeList
organizerOtherList =
rowElement.getElementsByTagName("event_organizer_other");
                                                Element
organizerOtherElement = (Element) organizerOtherList.item(0);
                                                NodeList ff
= organizerOtherElement.getChildNodes();

organizerOther = ((Node) ff.item(0)).getNodeValue().trim();

System.out.println(organizerOther);
                                        }
                                        catch(Exception e) {

System.out.println("Other organizer: null");
                                        }

                                        try {
                                                NodeList
organizerNameList =
rowElement.getElementsByTagName("event_organizer_name");
                                                Element
organizerNameElement = (Element) organizerNameList.item(0);
                                                NodeList
gg = organizerNameElement.getChildNodes();

organizerName = ((Node) gg.item(0)).getNodeValue().trim();
                                        }
                                        catch(Exception e) {

System.out.println("Organizer name: null");
                                        }

                                        try {
                                                NodeList
organizerEmailList =
rowElement.getElementsByTagName("event_organizer_email");
                                                Element
organizerEmailElement = (Element) organizerEmailList.item(0);
                                                NodeList
hh = organizerEmailElement.getChildNodes();

organizerEmail = ((Node) hh.item(0)).getNodeValue().trim();
                                        }
                                        catch(Exception e) {

System.out.println("Organizer email: null");
                                        }

                                        try {
                                                NodeList
organizerPhoneList =
rowElement.getElementsByTagName("event_organizer_phone");
                                                Element
organizerPhoneElement = (Element) organizerPhoneList.item(0);
                                                NodeList ii
= organizerPhoneElement.getChildNodes();

organizerPhone = ((Node) ii.item(0)).getNodeValue().trim();
                                        }
                                        catch(Exception e) {

System.out.println("Organizer phone: null");
                                        }

                                        try {
                                                NodeList
organizerDetailsList =
rowElement.getElementsByTagName("event_organizer_details");
                                                Element
organizerDetailsElement = (Element) organizerDetailsList.item(0);
                                                NodeList jj
= organizerDetailsElement.getChildNodes();

organizerDetails = ((Node) jj.item(0)).getNodeValue().trim();
                                        }
                                        catch(Exception e) {

System.out.println("Organizer details: null");
                                        }

                                        try {
//                                              File file =
new File(poster);
//                                              fileread =
new FileInputStream(file);
                                        }
                                        catch(Exception e) {

System.out.println("Event poster: null");
                                        }



                                        String query = "insert
into fiesta.table_event values (?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)";
                                        PreparedStatement
stmt = con.prepareStatement(query);

                                        stmt.setInt(1, eventid);
                                        stmt.setString(2,
eventname);
                                        stmt.setString(3,
eventdesc);
                                        stmt.setString(4, tags);
                                        stmt.setInt(5,
eventType);
                                        stmt.setString(6,
eventVenue);
                                        stmt.setInt(7,
maxParticipants);
```

```
                                                stmt.setString(8,
organizerCollege);
                                                stmt.setString(9,
organizerCampus);
                                                stmt.setString(10,
organizerName);
                                                stmt.setString(11,
organizerEmail);
                                                stmt.setString(12,
organizerPhone);
                                                stmt.setString(13,
organizerDetails);

stmt.setBinaryStream(14, fileread);
                                                stmt.setString(15,
organizerOther);
                                                stmt.setString(16,
registrationLink);
                                                stmt.setString(17,
startDateTime);
                                                stmt.setString(18,
endDateTime);

                                                try {
                                                        int q =
stmt.executeUpdate();

System.out.println("Inserted successfully! \n");
                                                }
                                                catch(Exception e) {

System.out.println("Either the data has already been inserted or its an
error during insertion!! \n");
                                                }

                                        }
                                }
                                con.close();
                        }
}
```

2) Table ID Counter

a) DTD

```
<!ELEMENT root (student_id, team_id, event_id, announcement_id,
contact_id, news_id)>
<!ELEMENT student_id (#PCDATA)>
<!ELEMENT team_id (#PCDATA)>
<!ELEMENT event_id (#PCDATA)>
<!ELEMENT announcement_id (#PCDATA)>
<!ELEMENT contact_id (#PCDATA)>
<!ELEMENT news_id (#PCDATA)>
```

b) XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
    <xs:element name="root" type="rootType" />

    <xs:complexType name="rootType">
        <xs:sequence>
            <xs:element name="student_id" type="IDType" />
            <xs:element name="team_id" type="IDType" />
            <xs:element name="event_id" type="IDType" />
            <xs:element name="announcement_id" type="IDType" />
            <xs:element name="contact_id" type="IDType" />
            <xs:element name="news_id" type="IDType" />
        </xs:sequence>
    </xs:complexType>

    <xs:simpleType name="IDType">
        <xs:restriction base="xs:integer">
            <xs:minInclusive value="0"></xs:minInclusive>
        </xs:restriction>
    </xs:simpleType>

</xs:schema>
```

c) XSL

```
<?xml version="1.0" encoding="UTF-8"?>

<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

    <xsl:template match="/">
        <html>
            <head>
                <link rel="stylesheet"
href="https://www.w3schools.com/w3css/4/w3.css" />
                <link rel="stylesheet" href="../css/tablecss.css" />
                <link rel="stylesheet" href="../css/style.css" />
                <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Sofia" />
            </head>
            <body>
                <center>
                    <h2>Table ID counter</h2>
                </center>
                <table class="table_props" id="table-title">
                    <tr id="table-heading">
                        <th>Student_ID</th>
                        <th>Team_ID</th>
                        <th>Event_ID</th>
                        <th>Announcement_ID</th>
                        <th>Contact_ID</th>
                        <th>News_ID</th>
                    </tr>
                    <xsl:for-each select="root">
                        <tr>
                            <td>
                                <xsl:value-of select="student_id" />
                            </td>
                            <td>
                                <xsl:value-of select="team_id" />
                            </td>
                            <td>
                                <xsl:value-of select="event_id" />
                            </td>
                            <td>
                                <xsl:value-of select="announcement_id" />
                            </td>
                            <td>
                                <xsl:value-of select="contact_id" />
                            </td>
                            <td>
                                <xsl:value-of select="news_id" />
                            </td>
                        </tr>
                    </xsl:for-each>
                </table>
            </body>
        </html>
    </xsl:template>
</xsl:stylesheet>
```

d) JDBC

```
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.DocumentBuilder;
import org.w3c.dom.Document;
import org.w3c.dom.NodeList;
import org.w3c.dom.Node;
import org.w3c.dom.Element;
import java.sql.*;
import java.io.File;

public class table_idcounter_xml_database_entry {
        public static void main(String[] args) throws Exception {
                Class.forName("com.mysql.cj.jdbc.Driver");
                Connection con =
DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/fiesta",
"root", "root");
                if (con != null)
                        System.out.println("Database
connected!");
                else
                        System.out.println("Database not
connected!");

    File id_counter_xmlfile = new File("./src/table_id_counter.xml");

                DocumentBuilderFactory docBuilderFactory =
DocumentBuilderFactory.newInstance();
                DocumentBuilder docBuilder =
docBuilderFactory.newDocumentBuilder();
                Document doc =
docBuilder.parse(id_counter_xmlfile);
                doc.getDocumentElement().normalize();
```

```
                        NodeList rows =
doc.getElementsByTagName("root");

                        for (int i = 0; i < rows.getLength(); i++) {
                                Node row = rows.item(i);

                                if (row.getNodeType() ==
Node.ELEMENT_NODE) {
                                        Element rowElement =
(Element) row;

                                        NodeList studentID =
rowElement.getElementsByTagName("student_id");
                                        Element
studentElement = (Element) studentID.item(0);
                                        NodeList textFNList =
studentElement.getChildNodes();
        String tmp = ((Node)
textFNList.item(0)).getNodeValue().trim();
        int student = Integer.parseInt(tmp);
NodeList teamID = rowElement.getElementsByTagName("team_id");
                                         Element teamElement
= (Element) teamID.item(0);
                                         NodeList textLNList =
teamElement.getChildNodes();
         tmp = ((Node) textLNList.item(0)).getNodeValue().trim();
         int team =  Integer.parseInt(tmp);
NodeList eventID =
rowElement.getElementsByTagName("event_id");
                                        Element eventElement
= (Element) eventID.item(0);
                                        NodeList textLNList1 =
eventElement.getChildNodes();
        tmp = ((Node) textLNList1.item(0)).getNodeValue().trim();
        int event = Integer.parseInt(tmp);

                                        NodeList
announcmentID =
rowElement.getElementsByTagName("announcement_id");
                                        Element
announcementElement = (Element) announcmentID.item(0);
                                        NodeList textLNList2 =
announcementElement.getChildNodes();
        tmp = ((Node) textLNList2.item(0)).getNodeValue().trim();
        int announcement = Integer.parseInt(tmp);

                                        NodeList contactID =
rowElement.getElementsByTagName("contact_id");
                                        Element
contactElement = (Element) contactID.item(0);
                                        NodeList textLNList3 =
contactElement.getChildNodes();
        tmp = ((Node) textLNList3.item(0)).getNodeValue().trim();
        int contact = Integer.parseInt(tmp);

                                        NodeList newsID =
rowElement.getElementsByTagName("news_id");
                                        Element newsElement
= (Element) newsID.item(0);
                                        NodeList textLNList4 =
newsElement.getChildNodes();
        tmp = ((Node) textLNList4.item(0)).getNodeValue().trim();
        int news = Integer.parseInt(tmp);

                                        String query = "insert
into fiesta.table_id_counter values (?,?,?,?,?,?)";
                                        PreparedStatement
stmt = con.prepareStatement(query);

stmt.setInt(1, student);
stmt.setInt(2, team);
stmt.setInt(3, event);
stmt.setInt(4, announcement);
stmt.setInt(5, contact);
stmt.setInt(6, news);
if (stmt.executeUpdate() == 1) {
System.out.println("Inserted successfully!");
                                         } else {

System.out.println("Error during insertion !!");
                                        }
                                }
                        }
                        con.close();
                }
}
```

Validation Test Cases

(A) Name: Gayathri E.
Roll No.: CB.EN.U4CSE17420

Field	Input Given	Success/Failure	Reason for Failure
student_id	Abc	Failure	Only numeric characters are allowed
student_id	-123	Failure	It should be greater than zero
student_id	0	Failure	1 is the minimum value allowed
student_id	10	Success	
student_achievement		Failure	Input cannot be null
student_achievement	BB event 1st place - Ref No: 1234567	Success	
student_achievement	BB event	Failure	Minimum length should be atleast 20
student_phone	9256453	Failure	Length should be 10
student_phone	987656453	Success	
student_phone	j087656453	Failure	Only numeric characters are allowed
student_phone	-0064532135	Failure	Number can't be negative
student_phone	0087656453	Failure	Number cannot start with 0
student_year_of_study	201230	Failure	Only 1 digit is allowed
student_year_of_study	1	Success	
student_year_of_study	6	Failure	Only 1-5 is allowed
student_about_myself		Failure	Input cannot be null
student_about_myself	Passionate Athlete working harder	Success	
student_about_myself	Passionate	Failure	Minimum length is 10 characters
student_school		Failure	Input cannot be null
student_socialmedia_1	amrita.com	Failure	accepts only http:// or https:// URIs
student_socialmedia_1	https://twitter.com/PatrickDempsey	Success	
student_socialmedia_2		Success	Input can be null
student_socialmedia_3	twitter.com/PatrickDempsey	Failure	accepts only http:// or https:// URIs

(B) Name: Prathyusha I
Roll No.: CB.EN.U4CSE17424

Field	Input given	Success/ Failure	Reason for failure
id	Cb1	Failure	Only integer values must be given.
id	1	Success	
id	1(duplicate value)	Failure	Only unique values must be given as it is primary key.
name	Prathyusha	Success	
name	12	Failure	Must be a string of characters but not numbers
gmail	prathyu24@gmail.com	Success	
gmail	Prathyu24gmail	Failure	Must be of corrected gmail format.
Password	Prathyu#234	Success	
student_id	14_cb	Failure	Must contain only digits.
Student_id	1	Success	
Primary_skill	Badminton	Success	
Ps_rating	rara	Failure	Must be an integer or a float number.
Ps_rating	3	Success	
Ps_rating	4.5	Success	
Team_Id	210	Success	
Team_id	blah	Failure	Must be numeric
team_id	210(duplicate value)	Failure	Only unique values must be given as it is primary key.
name	Igniters	Success	
name	Igniters123	Success	
name	142	Failure	Name must not contain numerical values alone.
Student_id	1	Success	
Student_id	5cbt	Failure	Must contain only numerical values

(D) Name: Sanjay Tharagesh R S
Roll No.: CB.EN.U4CSE17453

Field	Input Given	Success/Failure	Reason for Failure
contact_id	Abc	Failure	Only numeric characters are allowed
Contact_id	-123	Failure	It cannot be negative
Contact_id	0	Failure	Minimum allowed value is 1
Contact_id	12	Success	
Contact_name	Srishilesh	Success	
Contact_phone	-9087656453	Failure	Negative phone number is invalid
Contact_phone	9087656453	Success	
Contact_phone	j087656453	Failure	Only numeric characters are allowed
Contact_phone	6453	Failure	Length of phone number should be 10
Contact_phone	0087656453	Failure	Number cannot start with 0
Contact_email	xyz@gmail.com	Success	
Contact_email	qwe@.c	Failure	Invalid email format
announcement_id	Abcda	Failure	Only numeric characters are allowed
announcement_id	-12	Failure	It cannot be negative
announcement_id	0	Failure	Minimum allowed value is 1
announcement_id	14	Success	
event_id	Abcda	Failure	Only numeric characters are allowed
event_id	-12	Failure	It cannot be negative
event_id	0	Failure	Minimum allowed value is 1
event_id	14	Success	
team_id	Abcda	Failure	Only numeric characters are allowed
team_id	-12	Failure	It cannot be negative
team_id	0	Failure	Minimum allowed value is 1
team_id	14	Success	
position	2	Success	
Position	-1	Failure	Position cannot be negative
Position	0	Failure	Minimum position value is 1

Position	13	Failure	Maximum allowed value is 10

Name: Srishilesh P S

Roll No.: CB.EN.U4CSE17458

Field	Input Given	Success/Failure	Reason for Failure
event_id	xyz	Failure	Only numeric characters are allowed
event_id	-123	Failure	It cannot be negative
event_id	0	Success	Value start from 0
event_id	12	Success	
event_name	Anokha	Success	
event_desc	Annual cultural tech fest	Success	
event_tags	singing, dancing	Success	Automatically converts to lowercase
event_tags	GAMING, CoDIng	Success	Automatically converts to lowercase
event_start_datetime	2020-04-01T10:00:00	Success	
event_start_datetime	20-10-01T10:00:00	Failure	Wrong Timestamp format
event_organizer_phone	0087656453	Failure	Number cannot start with 0
event_organizer_email	xyz@gmail.com	Success	
event_organizer_email	qwe@.c	Failure	Invalid email format
event_registration_link	form.googl.co	Failure	Should match the URL format
event_organizer_other		Success	It can be null value
announcement_id	0	Success	Minimum allowed value is 0
announcement_id	14	Success	
event_id	Abcda	Failure	Only numeric characters are allowed
event_id	-12	Failure	It cannot be negative
team_id	Abcda	Failure	Only numeric characters are allowed
team_id	-12	Failure	It cannot be negative
team_id	14	Success	
news_id	2	Success	
news_id	-1	Failure	Position cannot be negative

news_id	0	Failure	Minimum position value is 1

Evaluation Sheet

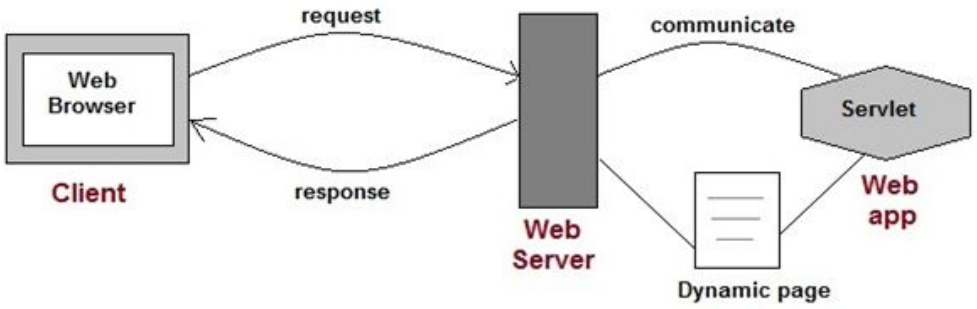
Roll No	Technology / Maximum Marks	Marks Awarded
CB.EN.U4CSE17420	XML(15) JDBC(15) VIVA(10) Total(50)	
CB.EN.U4CSE17424	XML(15) JDBC(15) VIVA(10) Total(50)	
CB.EN.U4CSE17445	XML(15) JDBC(15) VIVA(10) Total(50)	
CB.EN.U4CSE17453	XML(15) JDBC(15) VIVA(10) Total(50)	
CB.EN.U4CSE17458	XML(15) JDBC(15) VIVA(10) Total(50)	
	Project Documentation(10)	

TERM 3

Introduction to Servlets

Servlet Technology is used to create web applications. Servlet technology uses Java language to create web applications.

Web applications are helper applications that reside on a web server and build dynamic web pages. A dynamic page could be anything like a page that randomly chooses a picture to display or even a page that displays the current time.



As Servlet Technology uses Java, web applications made using Servlet are Secured, Scalable and Robust.

CGI (Common Gateway Interface)

Before Servlets, CGI(Common Gateway Interface) programming was used to create web applications. Here's how a CGI program works :

1. The user clicks a link that has a URL to a dynamic page instead of a static page.
2. The URL decides which CGI program to execute.
3. Web Servers run the CGI program in a separate OS shell. The shell includes the OS environment and the process to execute the code of the CGI program.
4. The CGI response is sent back to the Web Server, which wraps the response in an HTTP response and sends it back to the web browser.

Drawbacks of CGI programs

- High response time because CGI programs execute in their own OS shell.
- CGI is not scalable.
- CGI programs are not always secure or object-oriented.
    - It is Platform dependent.

Because of these disadvantages, developers started looking for better CGI solutions. And then Sun Micro-systems developed Servlet as a solution over traditional CGI technology.

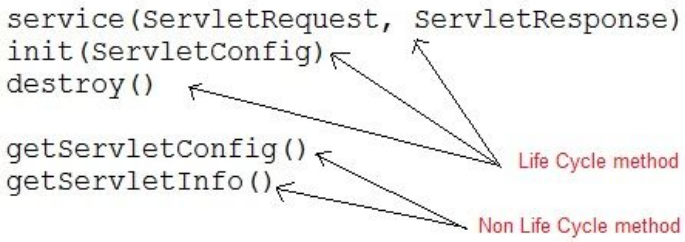
Advantages of using Servlets

- Less response time because each request runs in a separate thread.
- Servlets are scalable, robust, and object-oriented.
- Servlets are platform-independent.

Servlet API consists of two important packages that encapsulate all the important classes and interface, namely: java.servlet, javax.servlet.http

Servlet Interface

Servlet Interface provides five methods. Out of these five methods, three methods are Servlet life cycle methods and the rest two are non-life cycle methods.



How a Servlet Application works

A web container is responsible for managing the execution of servlets and JSP pages for Java EE applications. When a request comes in for a servlet, the server hands the request to the Web Container. Web Container is responsible for instantiating the servlet or creating a new thread to handle the request. It's the job of the Web Container to get the request and response to the servlet. The container creates multiple threads to process multiple requests to a single servlet. Servlets don't have a main() method. Web Container manages the life cycle of a Servlet instance.

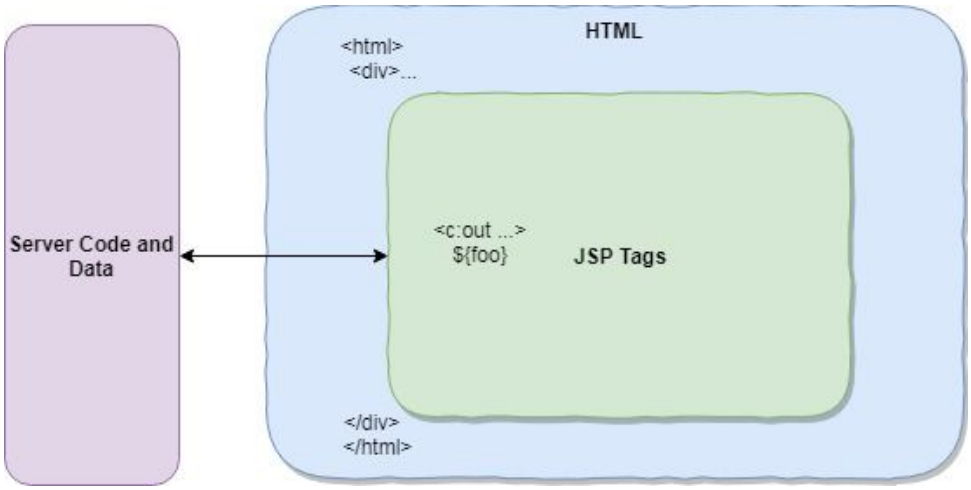
Applications of Servlet

1. Read the explicit data sent by the clients (browsers). This includes an HTML form on a Web page or it could also come from an applet or a custom HTTP client program.
2. Read the implicit HTTP request data sent by the clients (browsers). This includes cookies, media types and compression schemes the browser understands, and so forth.
3. Process the data and generate the results. This process may require talking to a database, executing an RMI or CORBA call, invoking a Web service, or computing the response directly.
4. Send the explicit data (i.e., the document) to the clients (browsers). This document can be sent in a variety of formats, including text (HTML or XML), binary (GIF images), Excel, etc.
5. Send the implicit HTTP response to the clients (browsers). This includes telling the browsers or other client's what type of document is being returned (e.g., HTML), setting cookies and caching parameters, and other such tasks.

Introduction to JSP

Java Server Pages (JSP) is a Java standard technology that enables you to write dynamic, data-driven pages for your Java web applications. JSP is built on top of the Java Servlet Specification. The two technologies typically work together, especially in older Java web applications. From a coding perspective, the most obvious difference between them is that with servlets you write Java code and then embed client-side markup (like HTML) into that code, whereas with JSP you start with the client-side script or markup, then embed JSP tags to connect your page to the Java backend.

A simple JSP page with extension *.jsp* consists of HTML markup embedded with JSP tags. When the file is processed on the server, the HTML is rendered as the application view, a web page. The embedded JSP tags will be used to call server-side code and data. The diagram below shows the interaction between HTML, JSP, and the web application server.



Why JSP over servlet?

- They are easy to maintain.
- No recompilation or redeployment is required.
- JSP has access to the entire API of JAVA .
- JSP are extended versions of Servlet.

Features of JSP:

- **Coding in JSP is easy**:- As it is just adding JAVA code to HTML/XML.
- **Reduction in the length of Code**:- In JSP we use action tags, custom tags, etc.
- **Connection to Database is easier**:-It is easier to connect a website to the database and allows you to read or write data easily to the database.
- **Make Interactive websites**:- In this, we can create dynamic web pages that help users to interact in a real-time environment.

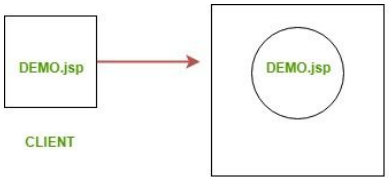
- **Portable, Powerful, flexible and easy to maintain**:- as these are browser and server independent.
- **No Redeployment and No Re-Compilation**:- It is dynamic, secure, and platform-independent so no need for re-compilation.
- **Extension to Servlet**:- as it has all features of servlets, implicit objects, and custom tags

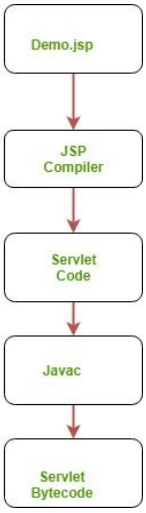
Basic JSP code:

```
<html>
<head><title>First JSP</title></head>
<body>
  <%
      double num = Math.random();
      if (num > 0.95) {
  %>
      <h2>You'll have a luck day!</h2><p>(<%= num %>)</p>
  <%
      } else {
  %>
      <h2>Well, life goes on ... </h2><p>(<%= num %>)</p>
  <%
      }
  %>
  <a href="<%= request.getRequestURI() %>"><h3>Try Again</h3></a>
</body>
</html>
```

Advantages:

1. It does not require advanced knowledge of JAVA
2. It is capable of handling exceptions
3. Easy to use and learn
4. It can tags which are easy to use and understand
5. Implicit objects are there which reduces the length of code
6. It is suitable for both JAVA and non-JAVA programmer





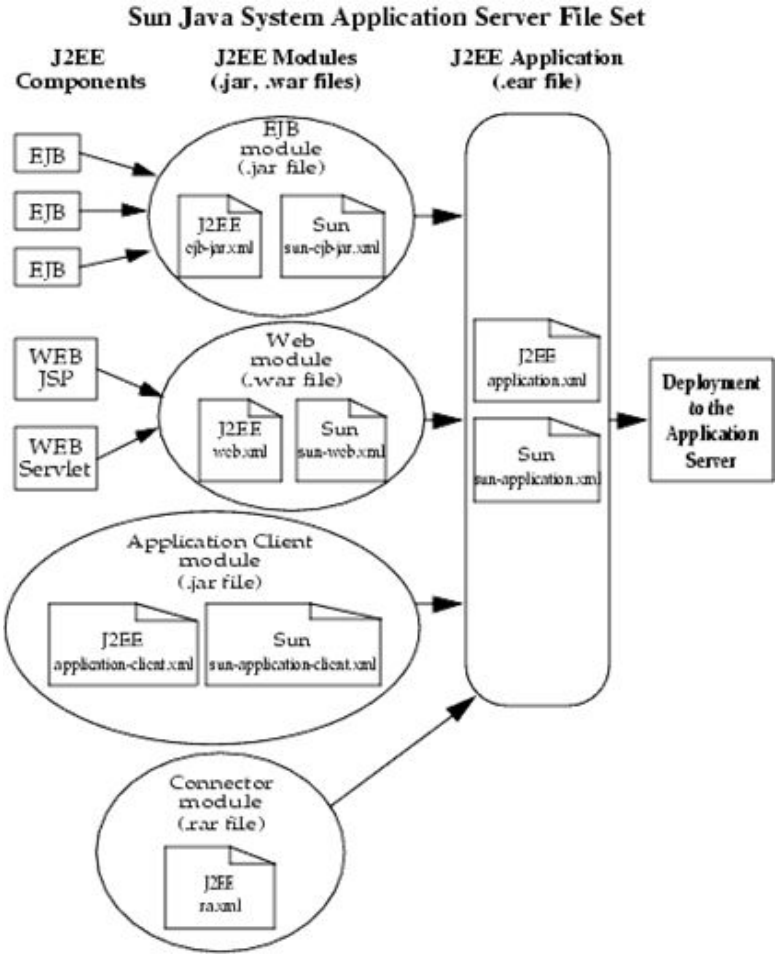
Disadvantages:

1. Difficult to debug for errors.
2. First-time access leads to wastage of time.
3. Its output is HTML which lacks features.

Steps in J2EE application deployment

A J2EE application is a logical collection of one or more J2EE modules tied together by application deployment descriptors. Components can be assembled at either the module or the application level. Components can also be deployed at either the module or the application level.

The following diagram illustrates how components are assembled into modules & then assembled into Sun Java System Application Server application EAR file ready for deployment.

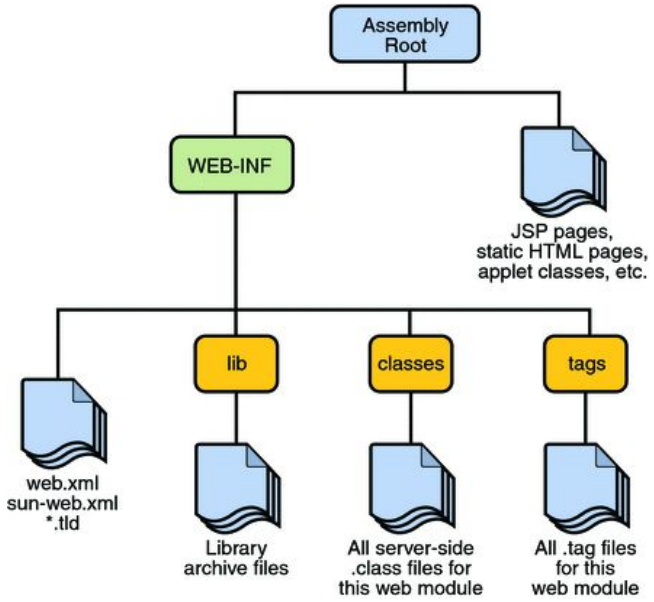


Deploy servlet in Apache's Tomcat Web Server

Steps for deploying a Servlet:

1. Create a web application
2. Install the Tomcat Web Server

3. Several subdirectories are automatically created under the "Tomcat Home" directory

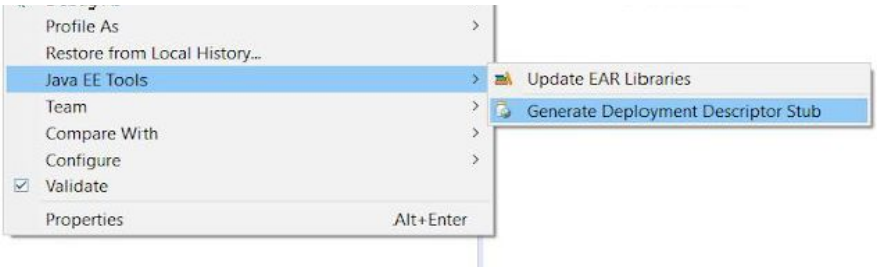


The jar files that are needed to deploy the particular web application are put under the "lib" directory which is under WEB-INF. "web.xml" contains the information needed to deploy the web components in the application. For example, if we take servlet it contains the definition of the servlet class and the URL mapping of the servlet etc JSP pages/HTML pages/other files that we need for a particular application.

4. Set the classpath to point to servlet-api.jar. Compile the servlet using the javac compiler

5. Deployment descriptor is an XML file(web.xml)

6. Define the servlet and give the servlet-mapping. To automatically create the web.xml file, select Java EE Tools -> Click Generate Development Descriptor Stub. This creates a web.xml which would be available in the WebContent/WEB-INF.



7. Run the application -> Run on the Tomcat server

Implementation

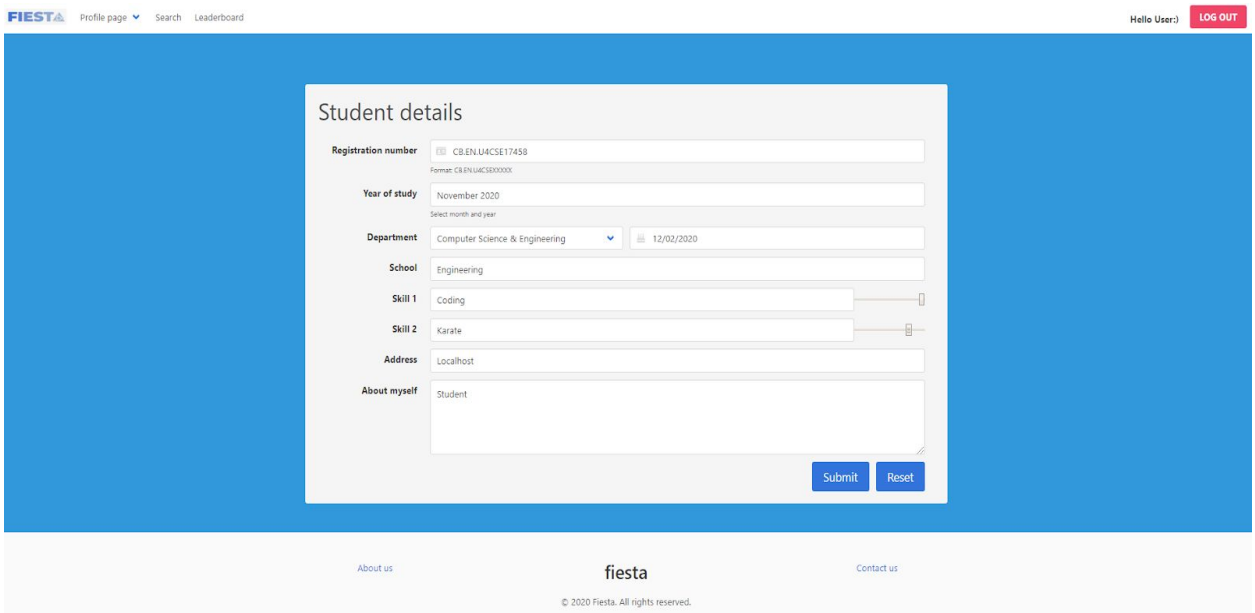
(A) Gayathri

Code

```
155     public static int getStudentID(HttpServletRequest request) {
156         Cookie cookie = null;
157         Cookie[] cookies = null;
158         // Get an array of Cookies associated with this domain
159         String student_id_str = "";
160         cookies = request.getCookies();
161         if( cookies != null ) {
162             // System.out.println("Found Cookies Name and Value");
163             for (int i = 0; i < cookies.length; i++) {
164                 cookie = cookies[i];
165                 if (cookie.getName().equals("student_id"))
166                     student_id_str = cookie.getValue();
167             }
168         }
169         else {
170             System.out.println("No cookies founds");
171         }
172         int student_id = 0;
173         if (student_id_str.equals(""))
174             student_id = 0;
175         else
176             student_id = Integer.parseInt(student_id_str);
177         return student_id;
178     }
```

```
118      protected void doPost(HttpServletRequest request, HttpServletResponse response)
119      throws ServletException, IOException {
120          String name = request.getParameter("name");
121          String regno = request.getParameter("reg_no");
122          String graduation = request.getParameter("graduation");
123          String dept = request.getParameter("dept");
124          String dob = request.getParameter("dob");
125          String school = request.getParameter("school");
126          String skill1 = request.getParameter("skill1");
127          String skill2 = request.getParameter("skill2");
128          String skill1_rating = request.getParameter("skill1_rating");
129          String skill2_rating = request.getParameter("skill2_rating");
130          String address = request.getParameter("address");
131          String aboutme = request.getParameter("aboutme");
132          int student_id = getStudentID(request);
133          try {
134              Class.forName("com.mysql.cj.jdbc.Driver");
135              Connection con = DriverManager.getConnection
136  ("jdbc:mysql://database-1.c4hq5iosxryf.us-east-1.rds.amazonaws.com/fiesta"
137   ,"admin","nithin_aakash");
138              String query =
139      "DELETE from fiesta.table_student_profile_personal where student_id=?";
140              PreparedStatement stmt = con.prepareStatement(query);
141              stmt.setInt(1, student_id);
142              int i = stmt.executeUpdate();
143              System.out.println("Deleted Personal Details successfully!");
144              con.close();
145          }
146          catch(Exception e) {
147              System.out.println(e);
148              System.exit(1);
149          }
150          insertIntoDatabase(student_id, name, regno, graduation, dept,
151  dob, school, skill1, skill2, skill1_rating, skill2_rating, address, aboutme);
152          response.sendRedirect(request.getContextPath() +"/studentHome");
153      }
```

Output



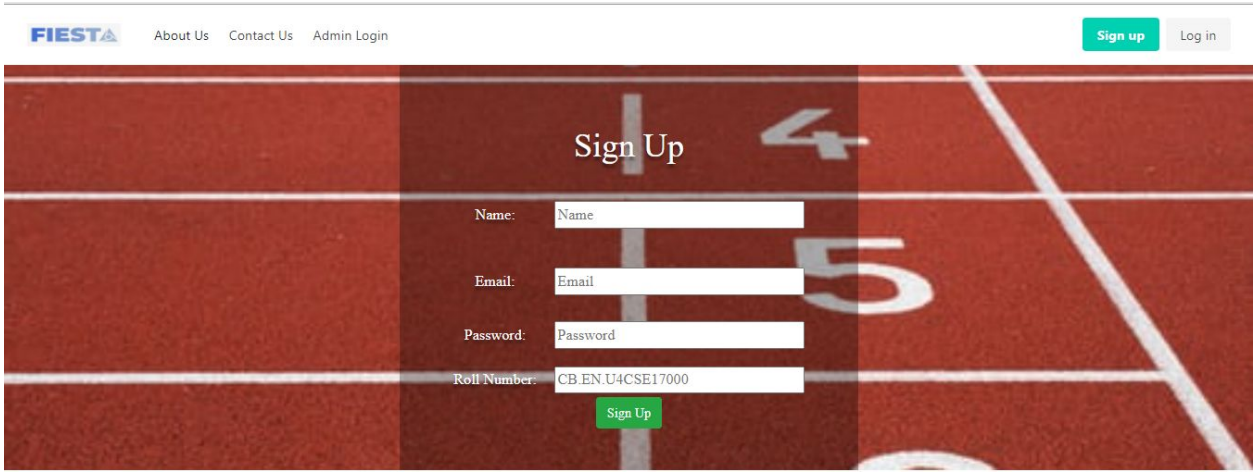
(B) Prathyusha

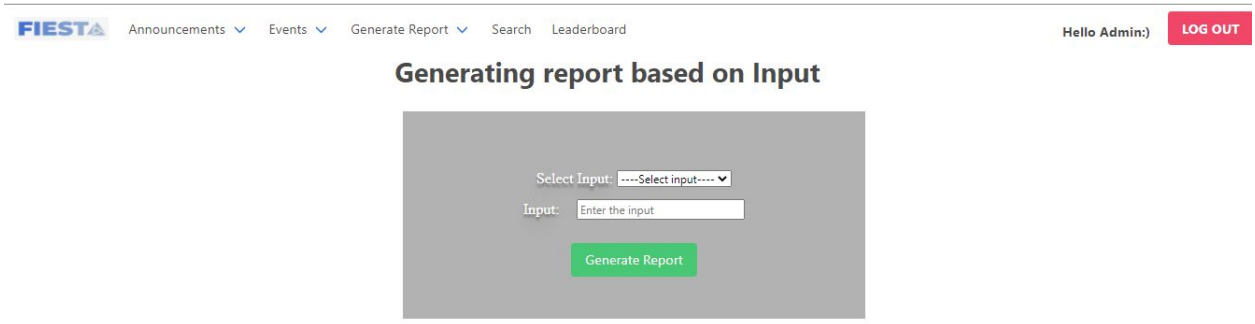
Code

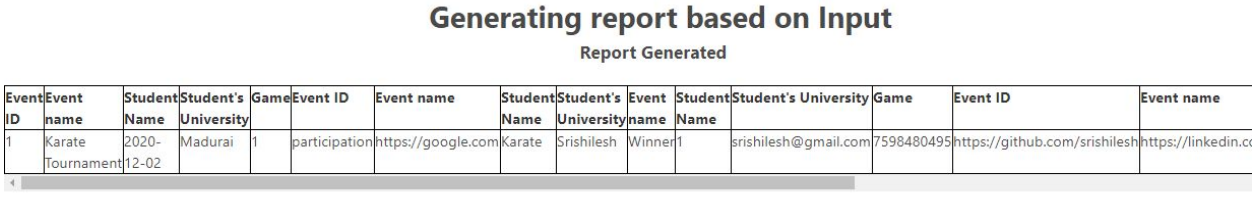
```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ervletException, IOException {
        // TODO Auto-generated method stub
//      doGet(request, response);
        String name = request.getParameter("name");
        String email = request.getParameter("email");
        String pword = request.getParameter("password");
        String roll = request.getParameter("roll");
        String student_id = studentIDTrigger() + "";
        try {
            if (insertIntoDatabase(student_id, name, email, pword, roll) == 1) {
                response.sendRedirect(request.getContextPath()+"/studentLogin");
            }
            else {
                response.sendRedirect(request.getContextPath()+"/registration");
            }

        }
        catch(Exception e) {
            System.out.println("Error: "+e);

        }
    }
```

```
            String input = request.getParameter("inp");

            ArrayList<ArrayList<String>> report = new ArrayList<ArrayList<String>>();
            try {
                Class.forName("com.mysql.cj.jdbc.Driver");
                Connection con = DriverManager.getConnection("jdbc:mysql://database-1.c4hq5iosxryf.us-east-
 1.rds.amazonaws.com/fiesta","admin","nithin_aakash");
                String select_announcements = "select distinct * from\r\n" +
                        "table_student_profile_achievements a\r\n" +
                        "inner join table_student_profile_contact c on a.student_id = c.student_id\r\n" +
                        "inner join table_student_profile_personal p on c.student_id = p.student_id\r\n" +
                        "where c.student_id=?;";
                PreparedStatement stmt = con.prepareStatement(select_announcements);
                stmt.setInt(1, Integer.parseInt(input));
                ResultSet rst = stmt.executeQuery();
                while (rst.next()) {

                    ArrayList<String> temp = new ArrayList<String>();
                    temp.add(rst.getInt(1)+"");
                    temp.add(rst.getString(2));
                    temp.add(rst.getDate(3)+"");
                    temp.add(rst.getString(4));
                    temp.add(rst.getInt(5)+"");
                    temp.add(rst.getString(6));
                    temp.add(rst.getString(7));
                    temp.add(rst.getString(8));
                    temp.add(rst.getString(9));
                    temp.add(rst.getString(10));
                    temp.add(rst.getInt(11)+"");
                    temp.add(rst.getString(12));
                    temp.add(rst.getString(13));
                    temp.add(rst.getString(14));
                    temp.add(rst.getString(15));
                    temp.add(rst.getInt(16)+"");
```

Output







Event ID	Event name	Student Name	Student's University	Game	Event ID	Event name	Student Name	Student's University	Event name	Student Name	Student's University	Game	Event ID	Event name
1	Karate Tournament	2020-12-02	Madurai	1	participation	https://google.com	Karate	Srishilesh	Winner	1	srishilesh@gmail.com	7598480495	https://github.com/srishilesh	https://linkedin.c

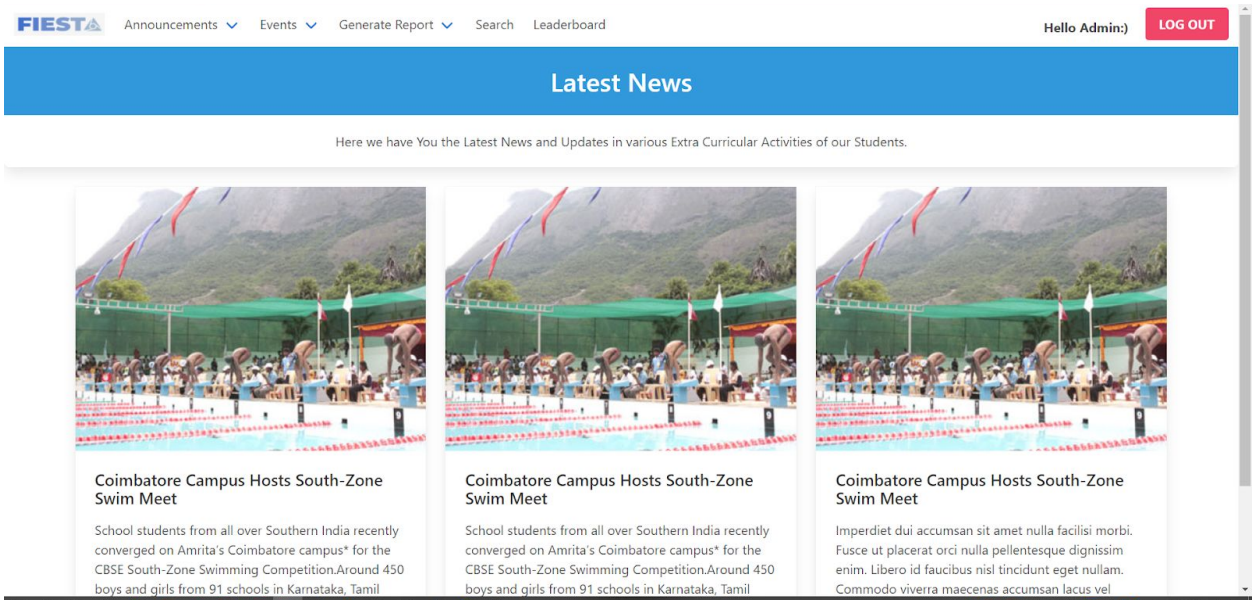
(C) Neeraj

Code

```
</script>
    <script>$(function(){
      $("#nav-placeholder").load("<%=request.getContextPath()%>/Navbar/adminPageNavbar.jsp");
    });</script>
</head>
<body>
    <div id="nav-placeholder"></div>
  <section class="hero is-info is-small">
      <div class="hero-body">
        <div class="container has-text-centered">
          <p class="title">
            Latest News
          </p>
        </div>
      </div>
    </section>
    <div class="box cta">
      <p class="has-text-centered">
        Here we have You the Latest News and Updates in various Extra Curricular Activities of our
Students.
      </p>
    </div>
    <section class="container">
      <div class="columns features" style="overflow-x: auto;">
        <%
            ArrayList<ArrayList<String>> listNews = (ArrayList<ArrayList<String>>)
request.getAttribute("latestNews");
```

```
            // print the information about every category of the list
            for(ArrayList<String> news : listNews) {%>
            <div class="column is-4">
              <div class="card is-dark">
                <div class="card-image">
                  <figure class="image is-4by3">
                    <img src="https://www.amrita.edu/sites/default/files/news-images/new/news-
events/images/k-oct/swim1.jpg">
                  </figure>
                </div>
                <div class="card-content">
                  <div class="content">
                    <h4><%=news.get(1)%></h4>
                    <p><%=news.get(2)%></p>
                    <a href=<%=news.get(3)%>><input type=button value='FULL DETAILS..'></a>
                  </div>
                </div>
              </div>
            </div>
```

Output



(D) Sanjay Tharagesh R S

```java
//Servlet implementation class UpdateWinners
@WebServlet("/updateWinners")
public class UpdateWinners extends HttpServlet {
    private static final long serialVersionUID = 1L;
    public UpdateWinners() {
        super();
        // TODO Auto-generated constructor stub
    }
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
        response.getWriter().append("Served at: ").append(request.getContextPath());
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            String path_to_db = "jdbc:mysql://database-1.c4hq5iosxryf.us-east-1.rds.amazonaws.com/fiesta";
            String username = "admin";
            String password = "nithin_aakash";
            Connection con = DriverManager.getConnection(path_to_db, username, password);
            ArrayList<ArrayList<String>> event_ids = new ArrayList<ArrayList<String>>();
            PreparedStatement stmt = con.prepareStatement("select event_id from fiesta.table_event");
            ResultSet rst = stmt.executeQuery();
            while (rst.next()) {
                ArrayList<String> tmp = new ArrayList<String>();
                tmp.add(rst.getInt(1)+"");
                event_ids.add(tmp);
            }
            con.close();
            request.setAttribute("event_ids", event_ids);
            System.out.println(event_ids);
            request.getRequestDispatcher("Events/update_winners.jsp").forward(request, response);
        }
        catch(Exception e) {
            System.out.println(e);
        }
    }
}
```

```java
//Servlet implementation class UpdateWinners
@WebServlet("/updateWinners")
public class UpdateWinners extends HttpServlet {
    private static final long serialVersionUID = 1L;
    public UpdateWinners() {
        super();
        // TODO Auto-generated constructor stub
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,
IOException {
        String event_id = request.getParameter("event_id");
        int count = Integer.parseInt(request.getParameter("count"));
        String winners[] = new String[count];
        for(Integer i=0; i<count; i++) {
            winners[i] = request.getParameter("message"+i.toString());
        }
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            String path_to_db = "jdbc:mysql://database-1.c4hq5iosxryf.us-east-1.rds.amazonaws.com/fiesta";
            String username = "admin";
            String password = "nithin_aakash";
            Connection con = DriverManager.getConnection(path_to_db, username, password);
            for(int i=0; i<count; i++) {
                PreparedStatement stmt = con.prepareStatement("INSERT INTO fiesta.table_event_winners
VALUES(?,?,?)");
                stmt.setInt(1, Integer.parseInt(event_id));
                stmt.setInt(2, i+1);
                stmt.setInt(3, Integer.parseInt(winners[i]));
                int rows_updated = stmt.executeUpdate();
                System.out.println(rows_updated + " records inserted");
            }
            for(int i=0; i<count; i++) {
                PreparedStatement stmt = con.prepareStatement("select student_id from fiesta.table_team_details
where team_id=?");
                stmt.setInt(1, Integer.parseInt(winners[i]));
                ResultSet rst = stmt.executeQuery();
                ArrayList<Integer> stu_ids = new ArrayList<Integer>();
                while (rst.next()) {
                    Integer stu_id = rst.getInt(1);
                    stu_ids.add(stu_id);
                }
                for(int j=0; j<stu_ids.size(); j++) {
                    stmt = con.prepareStatement("select student_id from fiesta.table_student_scores where
student_id=?");
                    stmt.setInt(1, stu_ids.get(j));
                    rst = stmt.executeQuery();
                    int rows_updated;
                    if(rst.next() == true)
                        stmt = con.prepareStatement("update fiesta.table_student_scores set
student_score=student_score+? where student_id=?");
                    else
                        stmt = con.prepareStatement("INSERT INTO fiesta.table_student_scores (`student_score`,
`student_id`) VALUES(?,?)");
                    stmt.setInt(1, 20-i-1);
                    stmt.setInt(2, stu_ids.get(j));
                    rows_updated = stmt.executeUpdate();
                    System.out.println(rows_updated + " records inserted");
                }
            }
            con.close();
            response.sendRedirect(request.getContextPath()+"/adminHome");
        }
        catch(Exception e) {
            System.out.println(e);
        }
    }
}
```

Upcoming events

Street Football

Play football in streets

Start Date & Time: 2020-11-12 19:49:00>
End Date & Time: 2020-11-30 19:49:00

Venue: Near AB2

Organizer College: Amrita Vishwa Vidyapeetham

Team Size: 11

Event ID: 6

Participation type: Team

View Event

Announcements!

FIFA event postponed

The FIFA gaming event has been postponed to October 29th.
Event ID:6

Go to Event 1

New Football match coming up!

Hey ballers! 29th Inter college football tournament is on the way. Follow the chanenl for more

Add new announcement!

Select Event ID

Title

NCP review postponed to 7th december

Message

NCP final project review will be conducted on 7th December. The panel list will be updated soon

Submit Cancel

Winners update form

Read the instructions carefully before submitting

- Enter the details correctly
- Enter only the Team ID
- Enter valid Team information
- Verify before submitting
- On submitting, the entered details will go public

Enter Event ID

12

Enter Number of Winners (Press Enter)

4

Fill Details

Place 1

11

Place 2

21

Place 3

211

Place 4

112

Submit

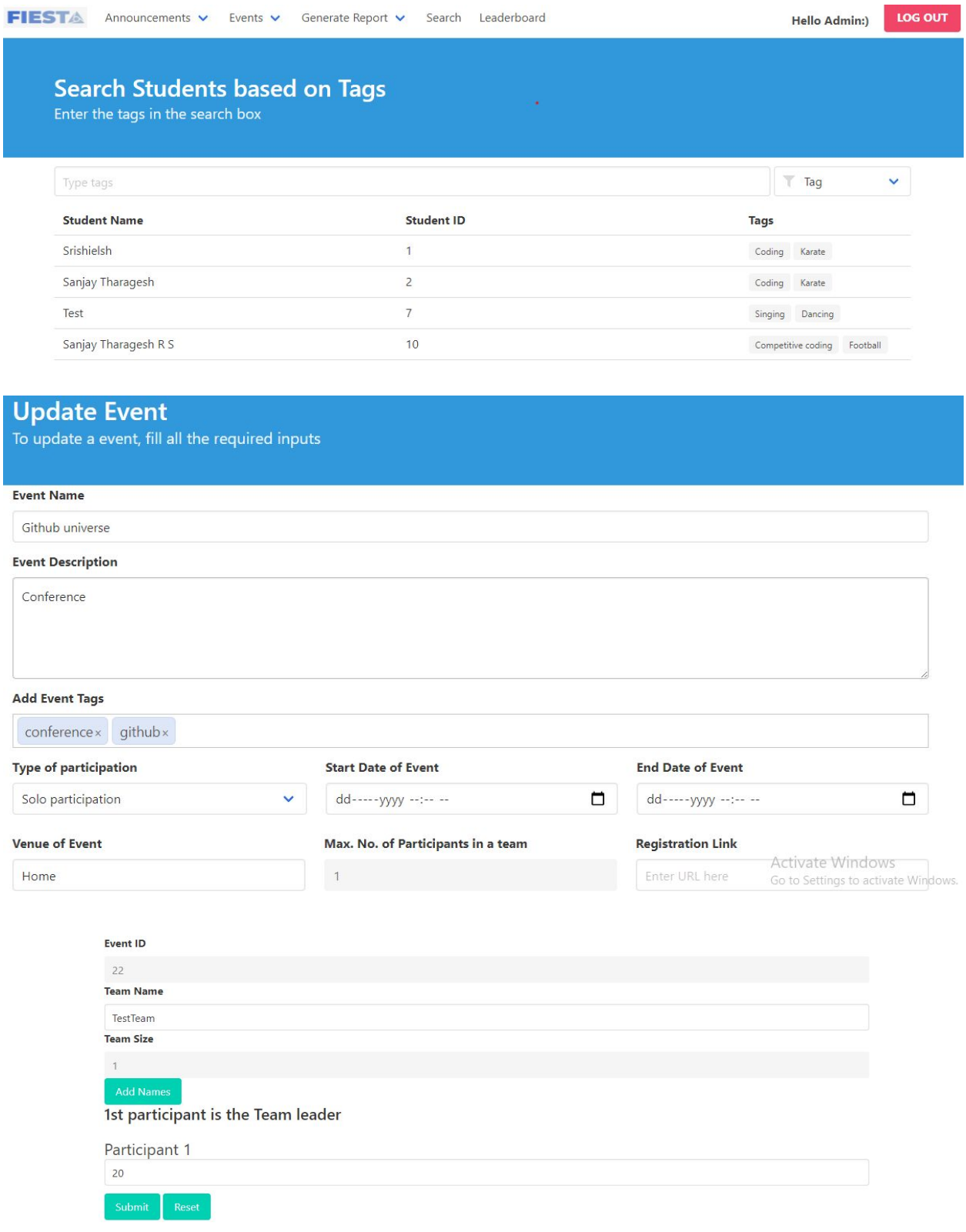
(E) Srishilesh P S

Code

```
@WebServlet("/addEvent")
public class CreateEventServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    public CreateEventServlet() {
        super();
    }
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        response.getWriter().append("Served at: ").append(request.getContextPath());

        response.sendRedirect("Events/create_event.jsp");
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
        try {
            String eventName = request.getParameter("event_name");
            String eventDesc = request.getParameter("event_desc");
            String eventTags = request.getParameter("event_tags");
            String eventTypeParticipation =
request.getParameter("event_type_participation");
            String eventStartDT = request.getParameter("event_start_datetime");
            String eventEndDT = request.getParameter("event_end_datetime");
            String eventVenue = request.getParameter("event_venue");
            String eventMaxParticipants = request.getParameter("event_max_participants");
            String eventRegistrationLink = request.getParameter("event_registration_link");
            String organizerCollege = request.getParameter("event_organizer_college");
            String organizerCampus = request.getParameter("event_organizer_campus");
            String organizerOther = request.getParameter("event_organizer_other");
            String organizerName = request.getParameter("event_organizer_name");
            String organizerEmail = request.getParameter("event_organizer_email");
            String organizerPhone = request.getParameter("event_organizer_phone");
            String organizerDetails = request.getParameter("event_organizer_details");
            // System.out.println(eventName+" "+eventDesc+" "+eventTags);
             int event_id = eventIDTrigger();
             insertIntoDatabase(event_id, eventName, eventDesc, eventTags,
eventTypeParticipation, eventStartDT, eventEndDT, eventVenue,
                    eventMaxParticipants, eventRegistrationLink, organizerCollege,
organizerCampus, organizerOther, organizerName, organizerEmail,
                    organizerPhone, organizerDetails);
        }
        catch(Exception e) {
            System.out.println("Error while retrieving data through POST method");
        }
        try {
            response.sendRedirect(request.getContextPath()+"/adminHome");
        }
        catch(Exception e) {
            System.out.println(e);
        }
    }
```

```
public static void insertIntoDatabase(int event_id, String eventName, String eventDesc,
String eventTags, String eventTypeParticipation,
            String eventStartDT, String eventEndDT, String eventVenue, String
eventMaxParticipants, String eventRegistrationLink,
            String organizerCollege, String organizerCampus, String organizerOther, String
organizerName, String organizerEmail,
            String organizerPhone, String organizerDetails) {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con = DriverManager.getConnection("jdbc:mysql://database-
1.c4hq5iosxryf.us-east-1.rds.amazonaws.com/fiesta","admin","nithin_aakash");

            System.out.println("-------------- ADMIN - CREATE EVENT ------------------");

            String query = "insert into fiesta.table_event values
(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)";
            PreparedStatement stmt = con.prepareStatement(query);

            int participationType = Integer.parseInt(eventTypeParticipation);
            int maxParticipants = 1;
            if (eventMaxParticipants == null)
                maxParticipants = 1;
            else
                maxParticipants = Integer.parseInt(eventMaxParticipants);

            stmt.setInt(1, event_id);
            stmt.setString(2, eventName);
            stmt.setString(3, eventDesc);
            stmt.setString(4, eventTags);
            stmt.setInt(5, participationType);
            stmt.setString(6, eventVenue);
            stmt.setInt(7, maxParticipants);
            stmt.setString(8, organizerCollege);
            stmt.setString(9, organizerCampus);
            stmt.setString(10, organizerName);
            stmt.setString(11, organizerEmail);
            stmt.setString(12, organizerPhone);
            stmt.setString(13, organizerDetails);
            stmt.setString(14, null);
            stmt.setString(15, organizerOther);
            stmt.setString(16, eventRegistrationLink);
            stmt.setString(17, eventStartDT);
            stmt.setString(18, eventEndDT);

            int i = stmt.executeUpdate();
            System.out.println("Inserted Event Details successfully!");
            con.close();
        }
        catch(Exception e) {
            System.out.println("Error during insertion: "+e);
            System.exit(1);
        }
    }
```

Output


FIESTA
Announcements
Events
Generate Report
Search
Leaderboard
Hello Admin:)
LOG OUT
Search Students based on Tags
Enter the tags in the search box
Type tags
Tag
Student Name
Student ID
Tags
Srishielsh
1
Coding
Karate
Sanjay Tharagesh
2
Coding
Karate
Test
7
Singing
Dancing
Sanjay Tharagesh R S
10
Competitive coding
Football
Update Event
To update a event, fill all the required inputs
Event Name
Github universe
Event Description
Conference
Add Event Tags
conference×
github×
Type of participation
Solo participation
Start Date of Event
dd-----yyyy --:-- --
End Date of Event
dd-----yyyy --:-- --
Venue of Event
Home
Max. No. of Participants in a team
1
Registration Link
Enter URL here
Activate Windows
Go to Settings to activate Windows.
Event ID
22
Team Name
TestTeam
Team Size
1
Add Names
1st participant is the Team leader
Participant 1
20
Submit
Reset

Validation Test Cases

(A) **Name: Gayathri E.**
Roll No.: CB.EN.U4CSE17420

Field	Input Given	Success/Failure	Reason for Failure
student_id	Abc	Failure	Only numeric characters are allowed
student_id	-123	Failure	It should be greater than zero
student_id	0	Failure	1 is the minimum value allowed
student_id	10	Success	
student_achievement		Failure	Input cannot be null
student_achievement	BB event 1st place - Ref No: 1234567	Success	
student_achievement	BB event	Failure	Minimum length should be atleast 20
student_phone	9256453	Failure	Length should be 10
student_phone	987656453	Success	
student_phone	j087656453	Failure	Only numeric characters are allowed
student_phone	-0064532135	Failure	Number can't be negative
student_phone	0087656453	Failure	Number cannot start with 0
student_year_of_study	201230	Failure	Only 1 digit is allowed
student_year_of_study	1	Success	
student_year_of_study	6	Failure	Only 1-5 is allowed
student_about_myself		Failure	Input cannot be null
student_about_myself	Passionate Athlete working harder	Success	
student_about_myself	Passionate	Failure	Minimum length is 10 characters
student_school		Failure	Input cannot be null
student_socialmedia_1	amrita.com	Failure	accepts only http:// or https:// URIs
student_socialmedia_1	https://twitter.com/PatrickDempsey	Success	
student_socialmedia_2		Success	Input can be null

student_socialmedia_3	twitter.com/PatrickDempsey	Failure	accepts only http:// or https:// URIs

(B) Name: Prathyusha I
Roll No.: CB.EN.U4CSE17424

Field	Input given	Success/ Failure	Reason for failure
id	Cb1	Failure	Only integer values must be given.
id	1	Success	
id	1(duplicate value)	Failure	Only unique values must be given as it is primary key.
name	Prathyusha	Success	
name	12	Failure	Must be a string of characters but not numbers
gmail	prathyu24@gmail.com	Success	
gmail	Prathyu24gmail	Failure	Must be of corrected gmail format.
Password	Prathyu#234	Success	
student_id	14_cb	Failure	Must contain only digits.
Student_id	1	Success	
Primary_skill	Badminton	Success	
Ps_rating	rara	Failure	Must be an integer or a float number.
Ps_rating	3	Success	
Ps_rating	4.5	Success	
Team_Id	210	Success	
Team_id	blah	Failure	Must be numeric
team_id	210(duplicate value)	Failure	Only unique values must be given as it is primary key.
name	Igniters	Success	
name	Igniters123	Success	
name	142	Failure	Name must not contain numerical values alone.
Student_id	1	Success	
Student_id	5cbt	Failure	Must contain only numerical values

(D) **Name: Sanjay Tharagesh R S**
Roll No.: CB.EN.U4CSE17453

Field	Input Given	Success/Failure	Reason for Failure
contact_id	Abc	Failure	Only numeric characters are allowed
Contact_id	-123	Failure	It cannot be negative
Contact_id	0	Failure	Minimum allowed value is 1
Contact_id	12	Success	
Contact_name	Srishilesh	Success	
Contact_phone	-9087656453	Failure	Negative phone number is invalid
Contact_phone	9087656453	Success	
Contact_phone	j087656453	Failure	Only numeric characters are allowed
Contact_phone	6453	Failure	Length of phone number should be 10
Contact_phone	0087656453	Failure	Number cannot start with 0
Contact_email	xyz@gmail.com	Success	
Contact_email	qwe@.c	Failure	Invalid email format
announcement_id	Abcda	Failure	Only numeric characters are allowed
announcement_id	-12	Failure	It cannot be negative
announcement_id	0	Failure	Minimum allowed value is 1
announcement_id	14	Success	
event_id	Abcda	Failure	Only numeric characters are allowed
event_id	-12	Failure	It cannot be negative
event_id	0	Failure	Minimum allowed value is 1
event_id	14	Success	
team_id	Abcda	Failure	Only numeric characters are allowed
team_id	-12	Failure	It cannot be negative
team_id	0	Failure	Minimum allowed value is 1
team_id	14	Success	
position	2	Success	
Position	-1	Failure	Position cannot be negative

Position	0	Failure	Minimum position value is 1
Position	13	Failure	Maximum allowed value is 10

(E) Name: Srishilesh P S
Roll No.: CB.EN.U4CSE17458

Field	Input Given	Success/Failure	Reason for Failure
event_id	xyz	Failure	Only numeric characters are allowed
event_id	-123	Failure	It cannot be negative
event_id	0	Success	Value start from 0
event_id	12	Success	
event_name	Anokha	Success	
event_desc	Annual cultural tech fest	Success	
event_tags	singing, dancing	Success	Automatically converts to lowercase
event_tags	GAMING, CoDIng	Success	Automatically converts to lowercase
event_start_datetime	2020-04-01T10:00:00	Success	
event_start_datetime	20-10-01T10:00:00	Failure	Wrong Timestamp format
event_organizer_phone	0087656453	Failure	Number cannot start with 0
event_organizer_email	xyz@gmail.com	Success	
event_organizer_email	qwe@.c	Failure	Invalid email format
event_registration_link	form.googl.co	Failure	Should match the URL format
event_organizer_other		Success	It can be null value
announcement_id	0	Success	Minimum allowed value is 0
announcement_id	14	Success	
event_id	Abcda	Failure	Only numeric characters are allowed
event_id	-12	Failure	It cannot be negative
team_id	Abcda	Failure	Only numeric characters are allowed
team_id	-12	Failure	It cannot be negative
team_id	14	Success	
news_id	2	Success	

news_id	-1	Failure	Position cannot be negative
news_id	0	Failure	Minimum position value is 1

Evaluation Sheet

Roll No	Technology	Max Marks	Marks Awarded
CB.EN.U4CSE17420	Servlet JSP Integration Report Total	15 15 10 10 50	
CB.EN.U4CSE17424	Servlet JSP Integration Report Total	15 15 10 10 50	
CB.EN.U4CSE17445	Servlet JSP Integration Report Total	15 15 10 10 50	
CB.EN.U4CSE17453	Servlet JSP Integration Report Total	15 15 10 10 50	
CB.EN.U4CSE17458	Servlet JSP Integration Report Total	15 15 10 10 50	