



INTERACT CLUB WEBSITE

*Informatics Practices – Investigatory Project
CHIREC International School*

~ Ashrita Madabhushi & Rhea Lakhotia

XII-A

ACKNOWLEDGEMENTS

I extend my heartfelt appreciation to Naseema Ma'am for her unwavering guidance and mentorship during the entire journey of my IP project. Her insightful feedback and encouragement were instrumental in shaping the project's success. I am truly grateful for her dedication to fostering a learning environment that goes beyond the classroom.

Furthermore, I would like to acknowledge the constant support and resources provided by our school, which played a crucial role in the realization of this project. The conducive atmosphere and the school's commitment to academic excellence have been pivotal in nurturing our intellectual growth.

A special mention of gratitude goes to the CBSE Board for designing a curriculum that encourages exploration and innovation. Their emphasis on practical application has been instrumental in shaping our understanding of Informatics Practices.

I would like to extend my deepest appreciation to my parents for their unwavering support throughout the entirety of my IP project. Their encouragement, understanding, and provision of resources have been indispensable in bringing this project to fruition. Their belief in my abilities has been a constant motivation, and I am truly grateful for their enduring support on this academic journey.

Last but not least, I want to express my appreciation to my peer who collaborated with me on this project. Their hard work, enthusiasm, and collaborative spirit significantly enriched the project, making it a truly collective effort. Together, we navigated challenges and celebrated achievements, creating a memorable and fulfilling experience.

INDEX

S.no	Title	Page No.
1.	Abstract	1
2.	Hardware Requirements	2
3.	Software Requirements	3 - 4
4.	Screens - Home Page	5 - 13
5.	Screens - About Us	14 - 22
6.	Screens - General Member Registration	23 - 24
7.	Screens - Member list	24 - 32
8.	Screens - Activity List	32 - 35
9.	Screens - Event List	35 - 37
10.	Screens - Login Page	38 - 40
11.	Screen - Board Member Registration	40
12.	Screen - Activity Registration	41 - 42
13.	Screen - Event Registration	42 - 43
14.	Graph Analysis Page and Code	44- 46
15.	Code for views.py	47 - 52
16.	Code urls.py	53 - 54
17.	Code for settings.py	55 - 59
18.	Code for models.py	60 - 64

S.no	Title	Page No.
19.	Code for forms.py	65
20.	MySQL Tables	66 - 69
21.	Future Extensions	70
22.	Bibliography	71

ABSTRACT

A website for the Interact Club of CHIREC International School.

Last year the board members of Interact Club, CHIREC faced a lot of issues. It was difficult and time-consuming to take registrations manually for over 200 members according to class and section. On top of that, it was hard to keep proper track of the members who had already participated in activities and those who had yet to do so. A few students came up to us saying that they had paid to become members but their details were not included in the interact club book.

This website enables us to provide a solution to these problems for the upcoming board members in the following years. It allows students to register themselves as members online. It enables the board to pull up reports according to various fields such as class, section, UID, payment(yes/no), activity(yes/no), etc. Additionally, it facilitates teachers to access the list of members which helps them while preparing the conduct certificate for the students at the end of the year and allows parents and other students to be informed about the happenings of the Interact club. Most importantly, it allows us to keep track of historical data for analysis based on the club.

HARDWARE REQUIREMENTS

Microsoft Windows 10

Processor: Intel® Core™ @ 1.60 GHz 1.80 GHz

RAM: 8.00 GB

Disk Space: 237 GB

Microsoft Windows 10 (x360 Convertible)

Processor: Intel® Core™ @ 2.10 GHz 2.59 GHz

RAM: 4.00 GB

Disk Space: 237 GB

SOFTWARE REQUIREMENTS

PYTHON

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991. It is used for web development (server-side), software development, mathematics, and system scripting. Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc). It has a simple syntax similar to the English language.

Python has a syntax that allows developers to write programs with fewer lines than some other programming languages. It runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick. It can be treated in a procedural way, an object-oriented way, or a functional way.

DJANGO

Django is a web application framework written in Python programming language. It is based on the MVT (Model View Template) design pattern. Django is very demanding due to its rapid development feature. It takes less time to build an application after collecting client requirements. This framework uses a famous tagline: The web framework for perfectionists with deadlines.

By using Django, we can build web applications in very little time. Django is designed in such a manner that it handles much of the configuration automatically, so we can focus on application development only.



MySQL

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed, and supported by MySQL AB, which is a Swedish company. It is released under an open-source license. So, you have nothing to pay to use it. MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages. It uses a standard form of the well-known SQL data language. Works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.

MySQL works very quickly and works well even with large data sets. It is very friendly to PHP, the most appreciated language for web development.

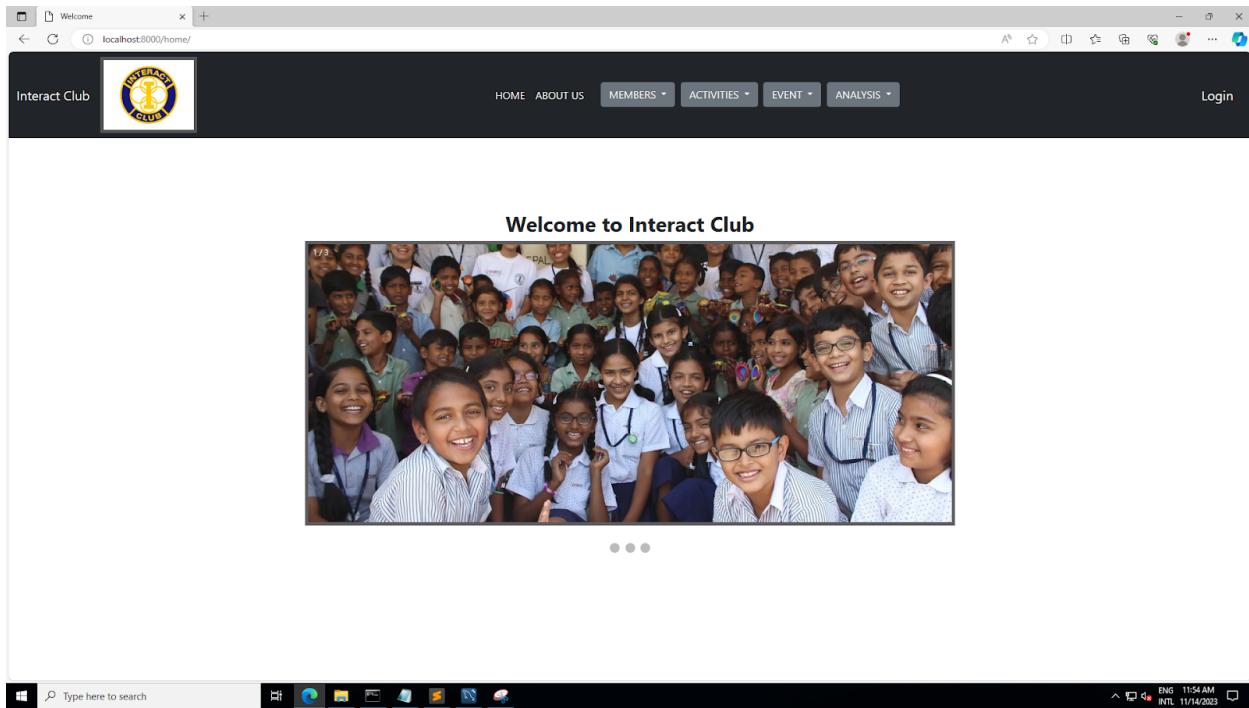
MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB). It is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.



SCREENS AND CODE

HOME PAGE

This is the Home Page of the Interact Club website. It has a slideshow of 3 pictures showcasing past activities that have already been conducted by the club.



BaseLayout.html

```
{% load static %}

<html>

    <head>

        <title>Welcome</title>

        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384--+0n0xVW2eSR5OomGNYDnhzAbDsOXxcvSN1TPprVMTNDbiYZCxYbO
Ol7+AMvyTG2x" crossorigin="anonymous">

    </head>

    <body onload="setupForm()">
```

```

<nav class="navbar navbar-expand-lg navbar-dark bg-dark shadow-5-strong">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Interact Club</a>
    
    class="img-responsive" alt="Responsive image">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <style>
      img {
        border: 5px solid #555;
      }
    </style>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
    data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent"
    aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse justify-content-center"
    id="navbarSupportedContent">
      <ul class="navbar-nav mb-2 mb-lg-0">
        <li class="nav-item">
          <a class="nav-link active" href="/home" aria-current="page" style="cursor:
pointer;">HOME</a>
        </li>
        <li class="nav-item">
          <a class="nav-link active" href="/aboutus" aria-current="page" style="cursor:
pointer;">ABOUT US</a>
        </li>
        <li class="nav-item">&ampnbsp&ampnbsp;
        </li>
        {% if user.is_authenticated %}

```

```

<li class="nav-item">
    <div class="dropdown">
        <button class="btn btn-secondary dropdown-toggle" type="button"
id="dropdownMenuButton1" data-bs-toggle="dropdown" aria-haspopup="true"
aria-expanded="false">
            ADMIN
        </button>
        <div class="dropdown-menu" aria-labelledby="dropdownMenuButton1">
            <a class="dropdown-item" href="/board/register">Member Registration</a>
            <a class="dropdown-item" href="/activity/register">Activity Registration</a>
            <a class="dropdown-item" href="/event/register">Event Registration</a>
        </div>
    </div>
</li>
{%
endif %}

<li class="nav-item">&nbsp;&nbsp;
</li>
<li class="nav-item">
    <div class="dropdown">
        <button class="btn btn-secondary dropdown-toggle" type="button"
id="dropdownMenuButton2" data-bs-toggle="dropdown" aria-haspopup="true"
aria-expanded="false">
            MEMBERS
        </button>
        <div class="dropdown-menu" aria-labelledby="dropdownMenuButton2">
            <a class="dropdown-item" href="/member/register">Member Registration</a>
            <a class="dropdown-item" href="/member/list">Member List</a>
        </div>
    </div>
</li>

```

```
<li class="nav-item">&nbsp;&nbsp;
</li>

<li class="nav-item">
  <div class="dropdown">
    <button class="btn btn-secondary dropdown-toggle" type="button"
id="dropdownMenuButton2" data-bs-toggle="dropdown" aria-haspopup="true"
aria-expanded="false"> ACTIVITIES
    </button>
    <div class="dropdown-menu" aria-labelledby="dropdownMenuButton2">
      <a class="dropdown-item" href="/activity/register">Activity Registration</a>
      <a class="dropdown-item" href="/activity/list">Activity List</a>
    </div>
  </li>

<li class="nav-item">&nbsp;&nbsp;
</li>

<li class="nav-item">
  <div class="dropdown">
    <button class="btn btn-secondary dropdown-toggle" type="button"
id="dropdownMenuButton2" data-bs-toggle="dropdown" aria-haspopup="true"
aria-expanded="false"> EVENT
    </button>
    <div class="dropdown-menu" aria-labelledby="dropdownMenuButton2">
      <a class="dropdown-item" href="/event/register">Event Registration</a>
      <a class="dropdown-item" href="/event/list">Event List</a>
    </div>
  </li>

<li class="nav-item">&nbsp;&nbsp;
</li>

<li class="nav-item">
  <div class="dropdown">
```

```

<button class="btn btn-secondary dropdown-toggle" type="button"
id="dropdownMenuButton2" data-bs-toggle="dropdown" aria-haspopup="true"
aria-expanded="false"> ANALYSIS
</button>
<div class="dropdown-menu" aria-labelledby="dropdownMenuButton2">
    <a class="dropdown-item" href="/graph/analysis">Graph</a>
</li>
</ul>
</div>
{% if user.is_authenticated %}
    &nbsp;&nbsp;<a class="navbar-brand">{{ user.username }}</a>
    &nbsp;&nbsp;<a class="navbar-brand" href="/board/logout">Logout</a>
    {% else %}
        &nbsp;&nbsp;<a class="navbar-brand" href="/board/login">Login</a>
    {% endif %}
</div>
</nav>
{% block content %}
{% endblock %}
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzbzo5smXKp4YfRvH+8abTE1Pi
6jizo" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/js/bootstrap.min.js"
integrity="sha384-Atwg2PkWv9vp0ygtn1JAojH0nYbwNJLPhwy0VbhoPwBhjQPR5VtM2+xf0
Uwh9KtT" crossorigin="anonymous"></script>
</body>
</html>

```

Home.html

```
{% extends 'BaseLayout.html' %}

{% block content %}

<br>
<br>
<br>

<p align="center">

<b>
<h2 style="text-align: center;">Welcome to Interact Club </h2></b>
<body style="background-color: #33475b">
>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
.mySlides {display: none;}
img {vertical-align: middle;}
/* Slideshow container */
.slideshow-container {
max-width: 1000px;
position: relative;
margin: auto;
}
/* Number text (1/3 etc) */
.numbertext {
color: #f2f2f2;
font-size: 12px;
padding: 8px 12px;
position: absolute;
}
```

```

top: 0;
}

/* The dots/bullets/indicators */

.dot {
height: 15px;
width: 15px;
margin: 0 2px;
background-color: #bbb;
border-radius: 50%;
display: inline-block;
transition: background-color 0.6s ease;
}

/* Fading animation */

.fade {
animation-name: fade;
animation-duration: 2s;
}

@keyframes fade {
from {opacity: .4}
to {opacity: 1}
}

/* On smaller screens, decrease text size */

@media only screen and (max-width: 300px) {
.text {font-size: 11px}
}

</style>
</head>
<body>

<div class="slideshow-container">

```

```

<div class="mySlides fade">
  <div class="numbertext">1 / 3</div>
  
</div>
<div class="mySlides fade">
  <div class="numbertext">2 / 3</div>
  
</div>
<div class="mySlides fade">
  <div class="numbertext">3 / 3</div>
  
</div>
</div>
<br>
<div style="text-align:center">
  <span class="dot"></span>
  <span class="dot"></span>
  <span class="dot"></span>
</div>
<script>
  let slideIndex = 0;
  showSlides();

function showSlides() {
  let i;
  let slides = document.getElementsByClassName("mySlides");

```

```
let dots = document.getElementsByClassName("dot");

for (i = 0; i < slides.length; i++) {
    slides[i].style.display = "none";
}

slideIndex++;

if (slideIndex > slides.length) {slideIndex = 1}

for (i = 0; i < dots.length; i++) {
    dots[i].className = dots[i].className.replace(" active", "");
}

slides[slideIndex-1].style.display = "block";
dots[slideIndex-1].className += " active";
setTimeout(showSlides, 2000); // Change image every 3 seconds

}

</script>
{%
  endblock %}

</body>
</html>
```

ABOUT US

This is the 'About Us' section of the website which gives a brief overview of CHIREC's community service initiatives via the club along with details of the Board Members of the Interact Club for the academic year.

The screenshot shows a Microsoft Edge browser window with the URL localhost:8000/aboutus/. The page has a dark header with the CHIREC logo, navigation links for HOME, ABOUT US, ADMIN, MEMBERS, ACTIVITIES, EVENT, ANALYSIS, and user information for 'chirecic' and 'Logout'. Below the header is a large photograph of a group of people, including students and adults, standing outdoors in front of a building. A caption below the photo reads 'About Us' and includes a detailed paragraph about CHIREC's mission to encourage compassionate global citizens through community service. Below this section is a heading 'Our Team' followed by four smaller photographs of individuals. The taskbar at the bottom shows various open applications like File Explorer, Microsoft Word, and a video player.

The screenshot shows the same Microsoft Edge browser window with the URL localhost:8000/aboutus/. The layout is identical to the first screenshot, featuring the 'About Us' section and the 'Our Team' section with four profile cards. Each card contains a small photo of a team member, their name, title, email address, and a 'Contact' button. The names and titles listed are: Ashrita Madabhushi (President, ashrita.madabhushi@gmail.com), Ankita (Vice-President, rhea.lakhotia@gmail.com), Nandini Nair (Secretary, nandini.nair@gmail.com), and Riddhima Futnani (Treasurer, riddhima.futnani@gmail.com). The taskbar at the bottom shows the same set of open applications as the previous screenshot.

aboutus.html

```
{% extends 'BaseLayout.html' %}

{% block content %}

{% load static %}

<!doctype html>

<html>

<head>

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <style>

        body {

            font-family: Arial, Helvetica, sans-serif;

            margin: 0;

        }

        html {

            box-sizing: border-box;

        }

        *, *:before, *:after {

            box-sizing: inherit;

        }

        .column {

            float: left;

            width: 33.3%;

            margin-bottom: 16px;

            padding: 0 8px;

        }

        .card {

            box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2);

            margin: 8px;

        }

    </style>

</head>

<body>
```

```
.about-section {  
padding: 50px;  
text-align: center;  
background-color: #474e5d;  
color: white;  
}  
.container {  
padding: 0 16px;  
}  
.container::after, .row::after {  
content: "";  
clear: both;  
display: table;  
}  
.title {  
color: grey;  
}  
.button {  
border: none;  
outline: 0;  
display: inline-block;  
padding: 8px;  
color: white;  
background-color: #000;  
text-align: center;  
cursor: pointer;  
width: 100%;  
}  
.button:hover {  
background-color: #555;
```

```
}

@media screen and (max-width: 650px) {

.column {
width: 100%;
display: block;
}

}

</style>

<meta name="viewport" content="width=device-width, initial-scale=1">
<style>

* {

box-sizing: border-box;
}

body {

font-family: Arial;
font-size: 17px;
}

.container {

position: relative;
max-width: 1000px ;
margin: 0 auto;
}

.container img {vertical-align: middle; }

.container .content {

position: absolute;
bottom: 0;
background: rgb(0, 0, 0); /* Fallback color */
background: rgba(0, 0, 0, 0.5); /* Black background with 0.5 opacity */
color: #f1f1f1;
max-width: 968px ;
```

```

padding: 20px;
}

</style>

</head>

<body>

<div class="container">

    &nbsp;&nbsp;

    <h3 >About Us</h3>

    <p>At CHIREC, community service is an integral part of our mission to encourage our students to be compassionate global citizens. In addition to developing the attitudes and skills necessary to make a difference in the world, CHIREC's community service programs aim to increase our students' awareness of real-world issues. We believe that social service helps develop a sense of community and responsibility. We give our students a platform to think critically about how they can contribute, and help them develop their ideas and knowledge to solve problems for a better world.</p>

    </div>

    </div>

    </div>

    &nbsp;&nbsp;

    <h2 style="text-align:center">Our Team</h2>

    <div class="row" style="padding-left: 30px" >

        <div class="column" >

            <div class="card">

            <div class="container">

```

```
<h3>Ashrita Madabhushi</h3>
<h6 class="title">President</h6>
<p>ashrita.madabhushi@gmail.com</p>
<p><button class="button">Contact</button></p>
</div>
</div>
</div>
<div class="column">
<div class="card">

<div class="container">
<h3>Ankita</h3>
<h6 class="title">Vice-President</h6>
<p>rhea.lakhotia@gmail.com</p>
<p><button class="button">Contact</button></p>
</div>
</div>
</div>
<div class="column">
<div class="card">

<div class="container">
<h3>Nandini Nair</h3>
<h6 class="title">Secretary</h6>
<p>nandini.nair@gmail.com</p>
<p><button class="button">Contact</button></p>
</div>
</div>
```

```

</div>
<div class="column">
  <div class="card">
    
    <div class="container">
      <h3>Riddhima Futnani</h3>
      <h6 class="title">Treasurer</h6>
      <p>riddhima.futnani@gmail.com</p>
      <p><button class="button">Contact</button></p>
    </div>
  </div>
</div>
</div>
</body>
</html>
<style>
  body {
    font-family: Arial, Helvetica, sans-serif;
    margin: 0;
  }
  html {
    box-sizing: border-box;
  }
  *, *:before, *:after {
    box-sizing: inherit;
  }
  .column {
    float: left;
    width: 24.5%;
  }
</style>

```

```
margin-bottom: 16px;  
padding: 0 8px;  
}  
.card {  
box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2);  
margin: 4px;  
}  
.about-section {  
padding: 50px;  
text-align: center;  
background-color: #474e5d;  
color: white;  
}  
.container {  
padding: 0 16px;  
}  
.container::after, .row::after {  
content: "";  
clear: both;  
display: table;  
}  
.title {  
color: grey;  
}  
.button {  
border: none;  
outline: 0;  
display: inline-block;  
padding: 8px;  
color: white;
```

```
background-color: #000;  
text-align: center;  
cursor: pointer;  
width: 100%;  
}  
.button:hover {  
background-color: #555;  
}  
@media screen and (max-width: 650px) {  
.column {  
width: 100%;  
display: block;  
}  
}  
  
</style>  
{% endblock %}
```

GENERAL MEMBER REGISTRATION

This section of the Interact Club website allows students from grade IX and above to register themselves as members. The details of the student who has registered are directly added to the Member List.

The screenshot shows a web browser window with the URL `localhost:8000/member/register/`. The page title is "New Member Registration". The header includes the "Interact Club" logo, navigation links for HOME, ABOUT US, MEMBERS, ACTIVITIES, EVENT, ANALYSIS, and a Login button. Below the header is a form with the following fields:

Name:	<input type="text"/>
School:	<input type="text"/>
Curriculum:	<input type="text"/>
Year:	<input type="text"/>
Uid:	<input type="text"/>
Grade:	<input type="text"/>
Section:	<input type="text"/>

At the bottom of the form is a "Save" button. The browser's address bar shows "Type here to search" and various icons. The system tray indicates the date and time as "11/14/2023 11:43 AM".

member_form.html

```
{% extends 'BaseLayout.html' %}

{% block content %}

{% load static %}

<br>
<h2 style="text-align: center;">New Member Registration</h2>
<br>
<br>
<form method="post" novalidate>

    {% csrf_token %}

    <p align="center">
```

```

<table
    class="table table-striped table-bordered w-auto">
    {{ form.as_table }}
</table>
</p>
<br>
<p style='text-align: center'><button type="submit">Save</button><br>
</form>
{% endblock %}
</body>

```

MEMBERS LIST

This is the ‘Member List’ section of the website. It allows the user to filter and pull up records of students as per their requirements (they may specify one or more parameters such as grade, section, etc.) through the ‘Filter’ button. The user may download the data as a ‘PDF’ and print the data using the ‘Print’ button. The ‘Reset Filter’ button allows the user to view the entire Member List.

The screenshot shows a web browser window displaying the 'Members List' page. The page has a dark header with the 'Interact Club' logo and navigation links for HOME, ABOUT US, MEMBERS, ACTIVITIES, EVENT, ANALYSIS, and Login. Below the header is a title 'Members List'. A search/filter section contains dropdown menus for 'Choose a Grade', 'Choose a Section', 'Choose a Year', 'Choose a Type', 'Activity', 'Payment', and an input field for 'Enter UID' with buttons for 'Filter', 'Reset Filter', and 'Print'. Below this is a table with columns: ID, NAME, UID, GRADE, SECTION, CURRICULUM, ROLE, TYPE, YEAR, SCHOOL, ACTIVITY, and PAYMENT. The table contains four rows of student data. At the bottom of the page is a Windows taskbar with a search bar, pinned icons for File Explorer, Edge, and File Manager, and system status indicators.

ID	NAME	UID	GRADE	SECTION	CURRICULUM	ROLE	TYPE	YEAR	SCHOOL	ACTIVITY	PAYMENT
1	Nikita Gaur	5492	9	B	CBSE	None	General	2023	Chirec	True	True
2	Hasini Gupta	4958	9	B	CBSE	None	General	2023	Chirec	True	True
3	Anjali Dandu	2173	9	D	CBSE	None	General	2023	Chirec	True	True
4	Drisha Reddy	2099	9	C	CBSE	None	General	2023	Chirec	True	True

ID	NAME	UID	GRADE	SECTION	CURRICULUM	ROLE	TYPE	YEAR	SCHOOL	ACTIVITY	PAYMENT
1	Nikita Gaur	5492	9	B	CBSE	None	General	2023	Chirec	True	True
2	Hasini Gupta	4958	9	B	CBSE	None	General	2023	Chirec	True	True
3	Anjali Dandu	2173	9	D	CBSE	None	General	2023	Chirec	True	True
4	Drisha Reddy	2099	9	C	CBSE	None	General	2023	Chirec	True	True
5	Vedant Noone	3755	9	A	CBSE	None	General	2023	Chirec	True	True
6	Parth Poddar	5273	10	C	CBSE	None	General	2023	Chirec	True	True
7	Mehak Gupta	3409	10	E	CBSE	None	General	2023	Chirec	True	True
8	Aanya Menon	2413	10	D	CBSE	None	General	2023	Chirec	True	True
9	Anjali Narayan	2415	10	F	CBSE	None	General	2023	Chirec	True	True
10	Nachiketa Prakash	4693	10	F	CBSE	None	General	2023	Chirec	True	True
11	Arnav Gaur	3766	11	C	CBSE	None	General	2023	Chirec	True	True
12	Lakshmi Mohana	4077	11	B	CBSE	None	General	2023	Chirec	True	True
13	Parth Singal	17	11	D	CBSE	None	General	2023	Chirec	True	True
14	Aryan Nethi	154	11	E	CBSE	None	General	2023	Chirec	True	True

member list.html

```
{% extends 'BaseLayout.html' %}

{% block content %}

{% load static %}

<script language='javascript'>

function printTable() {

    var divToPrint=document.getElementById("printableTable");

    newWin= window.open("");

    newWin.document.write(divToPrint.outerHTML);

    newWin.print();

    newWin.close();

}

function setupForm() {

    var urlParams = new URLSearchParams(window.location.search);
```

```

var grade = urlParams.get('grade');

var section = urlParams.get('section');

var type = urlParams.get('type');

var year = urlParams.get('year');

var uid = urlParams.get('uid');

var activity = urlParams.get('activity');

var payment = urlParams.get('payment');

//alert(activity);

if (grade != null)

    document.forms[0].elements['grade'].value = grade;

if (section != null)

    document.forms[0].elements['section'].value = section;

if (type != null)

    document.forms[0].elements['type'].value = type;

if (year != null)

    document.forms[0].elements['year'].value = year;

if (uid != null)

    document.forms[0].elements['uid'].value = uid;

if (activity != null)

    document.forms[0].elements['activity'].value = activity;

if (payment != null)

    document.forms[0].elements['payment'].value = payment;

}

</script>

<br>

{ % csrf_token % }

<form name="list" id="list" method="get" action="<% url 'member_changelist' %>">

```

```

<h2 style="text-align: center;">Members List</h2>
<br><br>
<p align="center">
<table class="table table-bordered w-auto">
<tr>
<th>
    <label for="grade">Choose a Grade:</label>
</th>
<td>
    <select name="grade" id="grade">
        <option value=""></option>
        {% for mem in grade %}
            <option value="{{ mem.id }}">{{ mem.name }}</option>
        {% endfor %}
    </select>
</td>
<th>
    <label for="section">Choose a Section:</label>
</th>
<td>
    <select name="section" id="section">
        <option value=""></option>
        {% for mem in section %}
            <option value="{{ mem.id }}">{{ mem.name }}</option>
        {% endfor %}
    </select>
</td>

```

```

<th>
    <label for="year">Choose a Year:</label>
</th>
<td>
    <select name="year" id="year">
        <option value=""></option>
        { % for mem in year %}
        <option value="{{ mem.id }}>{{ mem.name }}</option>
        { % endfor %}
    </select>
</td>
</tr>

<tr>
<th>
    <label for="type">Choose a Type:</label>
</th>
<td>
    <select name="type" id="type">
        <option value=""></option>
        { % for mem in type %}
        <option value="{{ mem.id }}>{{ mem.name }}</option>
        { % endfor %}
    </select>
</td>
<th>
    <label for="activity">Activity:</label>
</th>

```

```

<td>
    <select name="activity" id="activity">
        <option value=""></option>
        <option value="1">Yes</option>
        <option value="0">No</option>
    </select>
</td>
<th>
    <label for="payment">Payment:</label>
</th>
<td>
    <select name="payment" id="payment">
        <option value=""></option>
        <option value="1">Yes</option>
        <option value="0">No</option>
    </select>
</td>
</tr>
<tr>
    <th>
        <label for="uid">Enter UID:</label>
    </th>
    <td>
        <input type="text" name="uid" id="uid">
    </td>
    <td>
        <p><input type="submit" name="Filter" value="Filter"/>

```

```

</p>
</td>
<td>
<p>
    <input type="reset" name="Reset" value="Reset Filter"
    onclick="window.location.href='{{ url 'member_changelist' }}';">&nbsp;&nbsp;
    <button onclick="printTable();">Print</button>
</p>
</td>
</tr>
</thead>
</table>
</p>
<br>
<br>
<p align="center">
<table id="printableTable" class="table table-striped table-bordered w-auto">
<thead>
<style>
    tr:hover {background-color: lightgrey;}
</style>
</thead>
<tbody>
<tr>
    <th>ID</td>
    <th>NAME</td>
    <th>UID</td>
    <th>GRADE</td>

```

```

<th>SECTION</td>
<th>CURRICULUM</td>
<th>ROLE</td>
<th>TYPE</td>
<th>YEAR</td>
<th>SCHOOL</td>
<th>ACTIVITY</td>
<th>PAYMENT</td>
</tr>
{ % for mem in member_list %}
<tr>
{ % if user.is_authenticated %}
<td><a href="{{ mem.get_absolute_url }}/member/{{ mem.id }}">{{ mem.id }}</a></td>
{ % else %}
<td>{{ mem.id }}</td>
{ % endif %}
<td>{{ mem.name }}</td>
<td>{{ mem.uid }}</td>
<td>{{ mem.grade }}</td>
<td>{{ mem.section }}</td>
<td>{{ mem.curriculum }}</td>
<td>{{ mem.role }}</td>
<td>{{ mem.type }}</td>
<td>{{ mem.year }}</td>
<td>{{ mem.school }}</td>
<td>{{ mem.activity }}</td>
<td>{{ mem.payment }}</td>

```

```

</tr>
{% endfor %}

</table>

</p>

<br>

</form>

{% endblock %}

```

ACTIVITY LIST

This section of the website displays the details of the activities the Interact Club has conducted year-wise. The ‘Filter’ search box allows the user to search for a particular activity by entering details of that activity. The user may download the data as a ‘PDF’ and print the data using the ‘Print’ button.

ID	NAME	DESCRIPTION	YEAR	No Of MEMBERS
1	Independence Day	Orphanage visit	2022	4
2	Gratitude Day	Competitions for support staff of CHIREC	2022	30
3	Republic Day	Government school visit	2022	5
4	Diwali Diya Decoration	Orphanage visit	2022	35

activity_list.html

```

{% extends 'BaseLayout.html' %}

{% block content %}

```

```

{%- load static %}

<script language='javascript'>

    function printTable() {

        var divToPrint=document.getElementById("printableTable");

        newWin= window.open("");
        newWin.document.write(divToPrint.outerHTML);
        newWin.print();
        newWin.close();

    }

</script>
<br>

{%- csrf_token %}

<form name="list" id="list" method="get" action="{% url 'activity_changelist' %}">

    <h2 style="text-align: center;">Activity List</h2>

    <br><br>

    <p align="center"></p>

    <br>

    <br>

    <div class="container">

        <p align="center">

            <b>Filter</b>

            <input id="myInput" size="55" type="text" placeholder="Search..">&nbsp;&nbsp;<button
            onclick="printTable();">Print</button>

        </p>

        <br>

        <p align="center">

            <table id="printableTable" class="table table-striped table-bordered w-auto">

                <thead>

```

```

<style>tr:hover {background-color: lightgrey;} </style>
</thead>
<tr>
<th>ID</td>
<th>NAME</td>
<th>DESCRIPTION</td>
<th>YEAR</td>
<th>No Of MEMBERS</td>
</tr>
<tbody id="myTable">
{%
  for mem in activity_list %
}
<tr>
{%
  if user.is_authenticated %
}
<td><a href="{% mem.get_absolute_url %}/activity/{{ mem.id }}">{{ mem.id }}</a></td>
{%
  else %
}
<td>{{ mem.id }}</td>
{%
  endif %
}
<td>{{ mem.name }}</td>
<td>{{ mem.description }}</td>
<td>{{ mem.year }}</td>
<td>{{ mem.no_of_members }}</td>
</tr>
{%
  endfor %
}
</tbody>
</table>
</div>
</p>

```

```

<br>
</form>
{% endblock %}

```

EVENT LIST

This section of the website displays the details of the events the Interact Club has conducted year-wise. The ‘Filter’ search box allows the user to search for a particular event by entering details of that event. The user may download the data as a ‘PDF’ and print the data using the ‘Print’ button.

ID	NAME	DESCRIPTION	YEAR	No OF MEMBERS
1	Membership Drive	Registration of members - raised 11k	2023	240
2	Membership Drive	Registration of members - raised 1.1lakhs	2022	220

event_list.html

```

{%
    extends 'BaseLayout.html'
}

{%
    block content
}

{%
    load static
}

<script language='javascript'>

function printTable() {

    var divToPrint=document.getElementById("printableTable");

```

```

newWin= window.open("");
newWin.document.write(divToPrint.outerHTML);
newWin.print();
newWin.close();

}

</script>

<br>

{%- csrf_token %}

<form name="list" id="list" method="get" action="{% url 'event_changelist' %}">
    <h2 style="text-align: center;">Event List</h2>
    <br><br>
    <p align="center"></p>
    <br>
    <br>
    <div class="container">
        <p align="center">
            <b>Filter</b>
            <input id="myInput" size="55" type="text" placeholder="Search.."/>&nbsp;&nbsp;<button
            onclick="printTable();">Print</button>
        </p>
        <br>
        <p align="center">
            <table id="printableTable" class="table table-striped table-bordered w-auto">
                <thead>
                    <style>tr:hover {background-color: lightgrey;} </style>
                </thead>
                <tr>
                    <th>ID</td>
                    <th>NAME</td>
                    <th>DESCRIPTION</td>

```

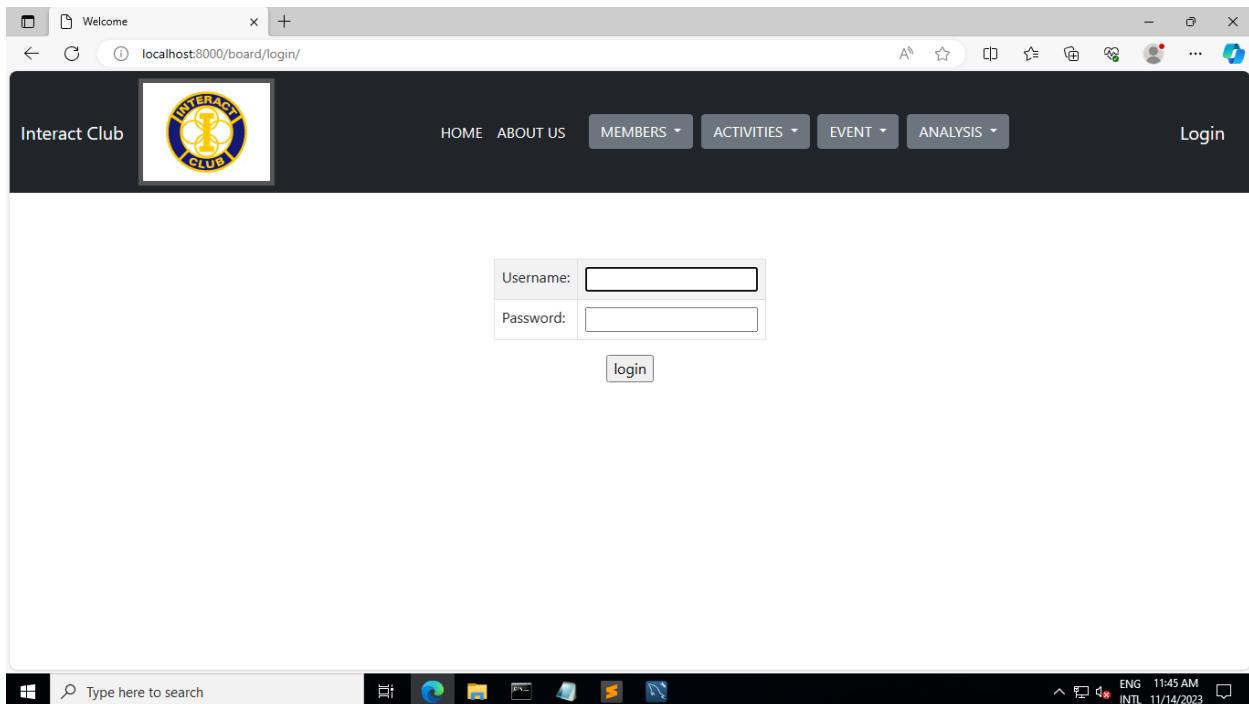
```

<th>YEAR</td>
<th>No Of MEMBERS</td>
</tr>
<tbody id="myTable">
    {% for mem in event_list %}
        {% if user.is_authenticated %}
            <td><a href="{{ mem.get_absolute_url }}/event/{{ mem.id }}">{{ mem.id }}</a></td>
        {% else %}
            <td>{{ mem.id }}</td>
        {% endif %}
            <td>{{ mem.name }}</td>
            <td>{{ mem.description }}</td>
            <td>{{ mem.year }}</td>
            <td>{{ mem.no_of_members }}</td>
        </tr>
    {% endfor %}
</tbody>
</table>
</div>
</p>
<br>
</form>
{% endblock %}

```

LOGIN PAGE

This is the authentication page which is used by board members to login to the site with their own account to register board members or to update any member, activity or event data. This is built using Django Authentication framework and the credentials for board members are obtained from admin console page (<http://localhost:8000/admin>) by logging in as super user with admin as username and admin as password.



login.html

```
{% extends 'BaseLayout.html' %}

{% block content %}

{% load static %}

{% if form.errors %}

<p>Your username and password didn't match. Please try again.</p>

{% endif %}

{% if next %}

{% if user.is_authenticated %}

<p>Your account doesn't have access to this page. To proceed,


please login with an account that has access.


```

```

</p>
{% else %}
<p>Please login to see this page.</p>
{% endif %}
{% endif %}

<br>
<br>
<br>

<form method="post" action="{% url 'login' %}">
    {% csrf_token %}
    <p align="center">
        <table class="table table-striped table-bordered w-auto">
            <tr>
                <td>{{ form.username.label_tag }}</td>
                <td>{{ form.username }}</td>
            </tr>
            <tr>
                <td>{{ form.password.label_tag }}</td>
                <td>{{ form.password }}</td>
            </tr>
        </table>
    </p>
    <p style='text-align: center'><input type="submit" value="login"></p>
    <input type="hidden" name="next" value="/">
</form>
{% endblock %}

```

logout.html

```

{% extends 'BaseLayout.html' %}

{% block content %}

```

```
{% load static %}

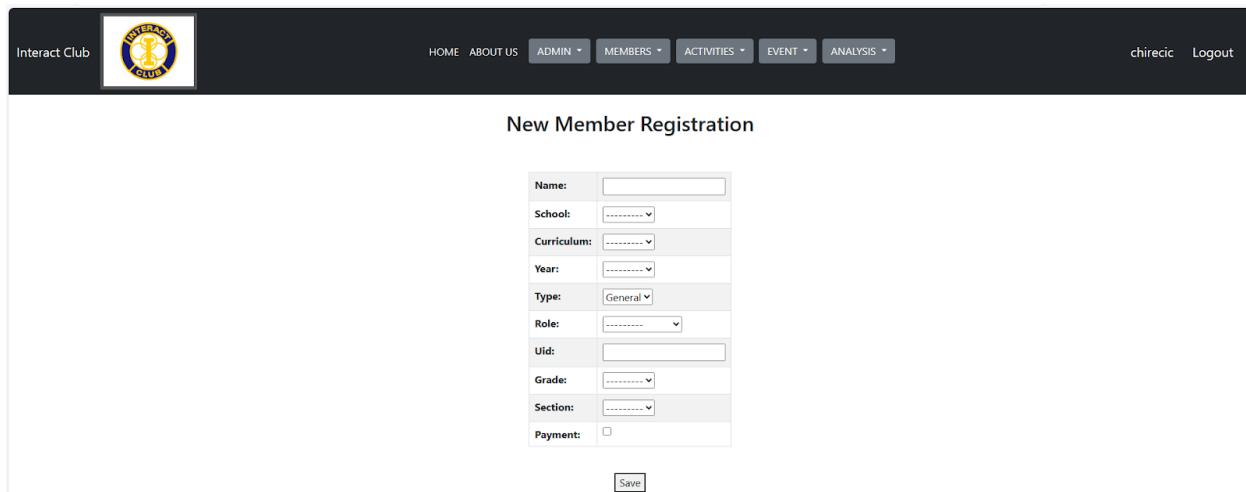
<p>Logged out!</p>

<a href="{% url 'login'%}">Click here to login again.</a>

{% endblock %}
```

BOARD MEMBER REGISTRATION

This section of the Interact Club website allows authenticated board members of Interact Club to register any board member. The Type on this page needs to be selected as ‘Board’. The details of the board member who has registered are directly added to the Member List.



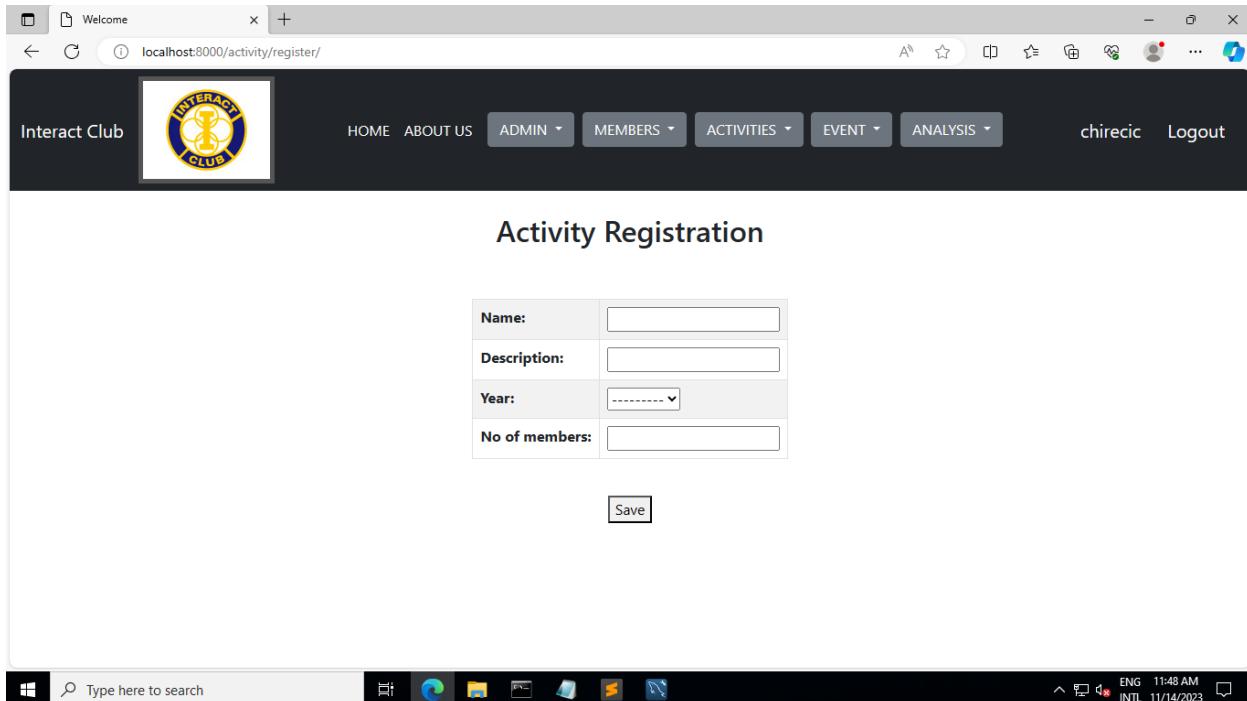
The screenshot shows a web application interface for 'New Member Registration'. At the top, there is a navigation bar with links for HOME, ABOUT US, ADMIN, MEMBERS, ACTIVITIES, EVENT, and ANALYSIS. On the far right of the navigation bar, it says 'chirecic' and 'Logout'. Below the navigation bar, the title 'New Member Registration' is centered. The main content area contains a form with the following fields:

Name:	<input type="text"/>
School:	<input type="text"/>
Curriculum:	<input type="text"/>
Year:	<input type="text"/>
Type:	<input type="text" value="General"/>
Role:	<input type="text"/>
Uid:	<input type="text"/>
Grade:	<input type="text"/>
Section:	<input type="text"/>
Payment:	<input type="checkbox"/>

At the bottom of the form is a 'Save' button.

ACTIVITY REGISTRATION

This section of the Interact Club website allows authenticated board members of Interact Club to register any activity conducted during that year.



activity_form.html

```
{% extends 'BaseLayout.html' %}

{% block content %}

{% load static %}

<br>
<h2 style="text-align: center;">Activity Registration</h2>
<br>
<br>
<form method="post" novalidate>

    {% csrf_token %}

    <p align="center">
        <table
            class="table table-striped table-bordered w-auto">
```

```

{{ form.as_table }}

</table>

</p>
<br>

<p style='text-align: center'><button type="submit">Save</button><br>
</form>

{% endblock %}

</body>

```

EVENT REGISTRATION

This section of the Interact Club website allows authenticated board members of Interact Club to register any fund raising event conducted during that year.

event_form.html

```

{% extends 'BaseLayout.html' %}

{% block content %}

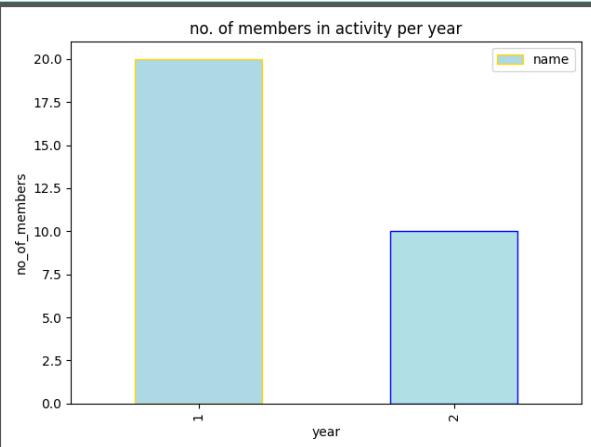
{% load static %}


```

```
<br>
<h2 style="text-align: center;">Event Registration</h2>
<br>
<br>
<form method="post" novalidate>
    {% csrf_token %}
    <p align="center">
        <table
            class="table table-striped table-bordered w-auto">
            {{ form.as_table }}
        </table>
    </p>
    <br>
    <p style='text-align: center'><button type="submit">Save</button><br>
</form>
{% endblock %}
</body>
```

GRAPHICAL ANALYSIS AND CODE

no. of members in activity per year



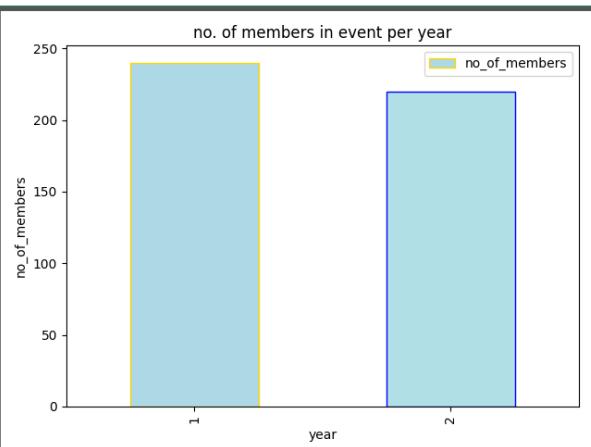
Graph Description

-This graph is a visual representation of the number of members who participated in an activity by Year

-Number of members participated in activities is along the x-axis and the year they have been participated is represented along the y-axis.

-This a bar graph in which different bars represent the number of members participated in activities by year.

no. of members in event per year



Graph Description

-This graph is a visual representation of the number of members who participated in events by Year

-Number of members participated in events is along the x-axis and the year they have been participated is represented along the y-axis.

-This a bar graph in which different bars represent the number of members participated in events by year.

graph.html

```
{% extends 'BaseLayout.html' %}

{% block content %}

{% load static %}

<!doctype html>

<html lang="en">

    <head>

        <link rel="stylesheet" href="{% static 'css/bootstrap.min.css' %}">
```

```

<script src="<% static 'js/bootstrap.min.js' %>"></script>
<title>Graph</title>
<style type="text/css"></style>
</head>
<div style="padding-left:50px class='container-fluid d-block'>
    <div class="container-fluid">
        <div class="row border-bottom border-success" style="padding:4px; color:rgb(15, 15, 160)">
            <h1 class="text-align: center"> no. of members in activity per year </h1>
        </div>
        <div class="row border-bottom">
            <div class="col-md-8 border-right" style="padding-left:100px; color:rgb(76, 22, 177)">
                <img src={{graphlib1}}>
            </div>
            <div class="col-md-4" style="color:rgb(177, 40, 22)">
                <h2> <b><i><u>Graph Description</u></i></b></h2>
                <br>
                <h4>
                    -This graph is a visual representation of the number of members who participated in an activity by Year<br><br>
                    -Number of members participated in activities is along the x-axis and the year they have been participated is represented along the y-axis.<br><br>
                    -This a bar graph in which different bars represent the number of members participated in activities by year.
                </h4>
            </div>
        </div>
    </div>
</div>

```

```

&nbsp;&nbsp;

<div style="padding-left:50px class='container-fluid d-block'>
  <div class="container-fluid">
    <div class="row border-bottom border-success" style="padding:4px; color:rgb(15, 15, 160)">
      <h1 class="text-align: center"> no. of members in event per year </h1>
    </div>
    <div class="row border-bottom">
      <div class="col-md-8 border-right" style="padding-left:100px; color:rgb(76, 22, 177)">
        <img src={{graphlib2}}>
      </div>
      <div class="col-md-4" style="color:rgb(177, 40, 22)">
        <h2> <b><i><u>Graph Description</u></i></b></h2>
        <br>
        <h4>
          -This graph is a visual representation of the number of members who participated in events by Year<br><br>
          -Number of members participated in events is along the x-axis and the year they have been participated is represented along the y-axis.<br><br>
          -This a bar graph in which different bars represent the number of members participated in events by year.
        </h4>
      </div>
    </div>
  </div>
</div>
{%
  endblock %
}
```

Code for views.py

```
from django.views.generic import ListView, CreateView, UpdateView
from django.urls import reverse_lazy
from .models import Member
from .models import Grade
from .models import Section
from .models import Year
from .models import Type
from .models import Activity
from .models import Event
from django.shortcuts import render
from django.contrib.auth.mixins import LoginRequiredMixin
from .forms import MemberUpdateForm
from .forms import EventUpdateForm
from .forms import ActivityUpdateForm
import pandas as pd
import matplotlib.pyplot as plt
import io
from io import BytesIO
import urllib, base64

def homefunction(request):
    return render(request,'home.html')

def aboutusfunction(request):
    return render(request,'aboutus.html')
```

```

def logoutfunction(request):
    return render(request,'logout.html')

class MemberCreateView(CreateView):
    model = Member
    fields = ('name', 'school','curriculum', 'year', 'uid', 'grade', 'section')
    success_url = reverse_lazy('member_changelist')

class BoardCreateView(LoginRequiredMixin, CreateView):
    model = Member
    login_url = '/board/login/'
    fields = ('name', 'school','curriculum', 'year', 'type','role', 'uid', 'grade', 'section','payment')
    success_url = reverse_lazy('member_changelist')

class MemberListView(ListView):
    model = Member
    context_object_name = "member_list"

    def get_context_data(self, **kwargs):
        context = super(MemberListView, self).get_context_data(**kwargs)
        filter_set = Member.objects.all()

        if (self.request.GET.get('grade') and self.request.GET.get('grade')!=""):
            grade_id_form = self.request.GET.get('grade')
            filter_set = filter_set.filter(grade_id=grade_id_form)

        if (self.request.GET.get('section') and self.request.GET.get('section')!=""):
```

```

section_id_form = self.request.GET.get('section')
filter_set = filter_set.filter(section_id=section_id_form)

if (self.request.GET.get('year') and self.request.GET.get('year')!=""):
    year_id_form = self.request.GET.get('year')
    filter_set = filter_set.filter(year_id=year_id_form)

if (self.request.GET.get('type') and self.request.GET.get('type')!=""):
    type_id_form = self.request.GET.get('type')
    filter_set = filter_set.filter(type_id=type_id_form)

if (self.request.GET.get('uid') and self.request.GET.get('uid')!=""):
    uid_id_form = self.request.GET.get('uid')
    filter_set = filter_set.filter(uid=uid_id_form)

if (self.request.GET.get('activity') and self.request.GET.get('activity')!=""):
    activity_form = self.request.GET.get('activity')
    filter_set = filter_set.filter(activity=activity_form)

if (self.request.GET.get('payment') and self.request.GET.get('payment')!=""):
    payment_form = self.request.GET.get('payment')
    filter_set = filter_set.filter(payment=payment_form)

context['member_list'] = filter_set
context['grade'] = Grade.objects.all()
context['section'] = Section.objects.all()
context['year'] = Year.objects.all()

```

```
context['type'] = Type.objects.all()

return context

class EventListView(ListView):
    model = Event
    context_object_name = "event_list"

class ActivityListView(ListView):
    model = Activity
    context_object_name = "activity_list"

class MemberUpdateView(LoginRequiredMixin, UpdateView):
    model = Member
    login_url = '/board/login/'
    fields = ('name', 'school','curriculum', 'type', 'role','year', 'uid', 'grade',
              'section','activity','payment')
    success_url = reverse_lazy('member_changelist')

class EventUpdateView(LoginRequiredMixin, UpdateView):
    model = Event
    login_url = '/board/login/'
    fields = ('name','description','year','no_of_members')
    success_url = reverse_lazy('event_changelist')

class ActivityUpdateView(LoginRequiredMixin, UpdateView):
    model = Activity
    login_url = '/board/login/'
```

```

fields = ('name','description','year','no_of_members')
success_url = reverse_lazy('activity_changelist')

class ActivityCreateView(LoginRequiredMixin, CreateView):
    model = Activity
    login_url = '/board/login/'

    fields = ('name','description','year','no_of_members')
    success_url = reverse_lazy('activity_changelist')

class EventCreateView(LoginRequiredMixin, CreateView):
    model = Event
    login_url = '/board/login/'

    fields = ('name','description','year','no_of_members')
    success_url = reverse_lazy('event_changelist')

def graph(request):

    filter_set = Member.objects.all()
    filter_set = filter_set.filter(activity=1)
    df2 = pd.DataFrame(filter_set.values())
    #print(df2)

    df2= df2.groupby(['year_id'], as_index=False)[["name"]].count()
    print (df2)

    df2.plot.bar(x="year_id", y="name", title="no. of members in activity per year",
    color=['lightblue', 'powderblue', 'palegreen', 'plum', 'mistyrose', 'lavender', 'gold',
    'silver'],edgecolor=['gold', 'blue', 'red', 'darkmagenta', 'deeppink', 'dodgerblue', 'green', 'red'])

    plt.xlabel('year')
    plt.ylabel('no_of_members')

```

```

plt.tight_layout()
df2.plot
fig=plt.gcf()
buf = BytesIO()
fig.savefig(buf, format='png', dpi=100)
buf.seek(0)
image_base64 = base64.b64encode(buf.read())
uri1 = 'data:image/png;base64,' + urllib.parse.quote(image_base64)
#print(uri)

df1 = pd.DataFrame(Event.objects.all().values())
#print(df1)

df1.plot.bar(x="year_id", y="no_of_members", title="no. of members in event per year",
color=['lightblue', 'powderblue', 'palegreen', 'plum', 'mistyrose', 'lavender', 'gold',
'silver'],edgecolor=['gold', 'blue', 'red', 'darkmagenta', 'deeppink', 'dodgerblue', 'green', 'red'])

plt.xlabel('year')
plt.ylabel('no_of_members')
plt.tight_layout()
df1.plot
fig=plt.gcf()
buf = BytesIO()
fig.savefig(buf, format='png', dpi=100)
buf.seek(0)
image_base64 = base64.b64encode(buf.read())
uri2 = 'data:image/png;base64,' + urllib.parse.quote(image_base64)
#print(uri)

return render(request,"graph.html",{"graphlib1": uri1, "graphlib2": uri2})

```

Code for urls.py

```
"""
```

URL configuration for InteractClub project.

The `urlpatterns` list routes URLs to views. For more information please see:

<https://docs.djangoproject.com/en/4.2/topics/http/urls/>

Examples:

Function views

1. Add an import: from my_app import views
2. Add a URL to urlpatterns: path("", views.home, name='home')

Class-based views

1. Add an import: from other_app.views import Home
2. Add a URL to urlpatterns: path("", Home.as_view(), name='home')

Including another URLconf

1. Import the include() function: from django.urls import include, path
2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))

```
"""
```

```
from django.contrib import admin
from django.urls import path, include
from members import views
```

```
urlpatterns = [
    path('admin/', admin.site.urls),
    path("",views.homefunction,name='home'),
    path('logout/',views.logoutfunction,name='logout'),
    path('home/',views.homefunction,name='home'),
```

```
path('aboutus/',views.aboutusfunction,name='aboutus'),  
path('board/', include('django.contrib.auth.urls')),  
path('member/list/', views.MemberListView.as_view(), name='member_changelist'),  
path('member/register/', views.MemberCreateView.as_view(), name='member_add'),  
path('board/register/', views.BoardCreateView.as_view(), name='board_add'),  
path('member/<int:pk>/', views.MemberUpdateView.as_view(), name='member_change'),  
path('event/register/', views.EventCreateView.as_view(), name='event_add'),  
path('activity/register/', views.ActivityCreateView.as_view(), name='activity_add'),  
path('event/list/', views.EventListView.as_view(), name='event_changelist'),  
path('activity/list/', views.ActivityListView.as_view(), name='activity_changelist'),  
path('event/<int:pk>/', views.EventUpdateView.as_view(), name='event_change'),  
path('activity/<int:pk>/', views.ActivityUpdateView.as_view(), name='activity_change'),  
path('graph/analysis/', views.graph, name='graph'),
```

]

Code for settings.py

```
"""
```

Django settings for InteractClub project.

Generated by 'django-admin startproject' using Django 4.2.1.

For more information on this file, see

<https://docs.djangoproject.com/en/4.2/topics/settings/>

For the full list of settings and their values, see

<https://docs.djangoproject.com/en/4.2/ref/settings/>

```
"""
```

```
from pathlib import Path
```

```
import os # needed by code below
```

```
# Build paths inside the project like this: BASE_DIR / 'subdir'.
```

```
BASE_DIR = Path(__file__).resolve().parent.parent
```

```
# Quick-start development settings - unsuitable for production
```

```
# See https://docs.djangoproject.com/en/4.2/howto/deployment/checklist/
```

```
# SECURITY WARNING: keep the secret key used in production secret!
```

```
SECRET_KEY =
```

```
'django-insecure-y5dr&uc_g@36&s4)9sp8cgpzu4eef^p*w@exh2q(!*@*8l9!t7'
```

```
# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = ['ec2-43-205-81-93.ap-south-1.compute.amazonaws.com','localhost']

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'members.apps.MembersConfig',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]
```

```
ROOT_URLCONF = 'InteractClub.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'templates')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'InteractClub.wsgi.application'
```

```
# Database
# https://docs.djangoproject.com/en/4.2/ref/settings/#databases
```

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'ic',
```

```

'USER': 'root',
'PASSWORD': 'admin',
'HOST': 'localhost',
'PORT': '3306',
'OPTIONS': {
    'init_command': "SET sql_mode='STRICT_TRANS_TABLES'"
}
}

}

}

# Password validation
# https://docs.djangoproject.com/en/4.2/ref/settings/#auth-passwordValidators

AUTH_PASSWORD_VALIDATORS = [
{
    'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
},
{
    'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
},
{
    'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
},
{
    'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
},
]

```

```
# Internationalization
# https://docs.djangoproject.com/en/4.2/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.2/howto/static-files/

STATIC_URL = 'static/'

LOGIN_REDIRECT_URL = '/'

LOGOUT_REDIRECT_URL = "/logout"

# Default primary key field type
# https://docs.djangoproject.com/en/4.2/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

Code for models.py

```
from django.db import models

# from django.db import models

class Section(models.Model):

    class Meta:
        db_table = "section"

    def __str__(self):
        return self.name

    name = models.CharField(max_length=45)
    description = models.CharField(max_length=45)

class Curriculum(models.Model):

    class Meta:
        db_table = "curriculum"

    def __str__(self):
        return self.name

    name = models.CharField(max_length=45)

class Grade(models.Model):

    class Meta:
```

```
db_table = "grade"

def __str__(self):
    return self.name

name = models.CharField(max_length=45)

class School(models.Model):
    class Meta:
        db_table = "school"

    def __str__(self):
        return self.name

    name = models.CharField(max_length=45)

class Role(models.Model):
    class Meta:
        db_table = "role"

    def __str__(self):
        return self.name

    name = models.CharField(max_length=45)
```

```
class Type(models.Model):
    class Meta:
        db_table = "type"

    def __str__(self):
        return self.name

    name = models.CharField(max_length=45)
```

```
class Year(models.Model):
    class Meta:
        db_table = "year"

    def __str__(self):
        return self.name

    name = models.CharField(max_length=45)
    enabled = models.BooleanField()
```

```
class Member(models.Model):
    class Meta:
        db_table = "member"
        constraints = [
```

```

        models.UniqueConstraint(fields=['uid', 'year'], name='unique_index'),
    ]

def __str__(self):
    return self.name

name = models.CharField(max_length=100)
uid = models.IntegerField()
section = models.ForeignKey(Section, on_delete=models.RESTRICT, blank=False,
null=False)
role = models.ForeignKey(Role, on_delete=models.SET_NULL, blank=True, null=True)
activity = models.BooleanField(default=False)
grade = models.ForeignKey(Grade, on_delete=models.RESTRICT, blank=False, null=False)
payment = models.BooleanField(default=False)
curriculum = models.ForeignKey(Curriculum, on_delete=models.RESTRICT, blank=False,
null=False)
year = models.ForeignKey(Year, on_delete=models.RESTRICT, blank=False, null=False)
school = models.ForeignKey(School, on_delete=models.RESTRICT, blank=False,
null=False)
type = models.ForeignKey(Type, on_delete=models.RESTRICT, blank=False, null=False,
default=1)

class Event(models.Model):
    class Meta:
        db_table = "event"

    def __str__(self):
        return self.name

```

```
name = models.CharField(max_length=45)
description= models.CharField(max_length=45)
year = models.ForeignKey(Year, on_delete=models.RESTRICT, blank=False, null=False,
limit_choices_to={"enabled" : True})
no_of_members = models.IntegerField()

class Activity(models.Model):
    class Meta:
        db_table = "activity"

    def __str__(self):
        return self.name

    name = models.CharField(max_length=45)
    description= models.CharField(max_length=45)
    year = models.ForeignKey(Year, on_delete=models.RESTRICT, blank=False, null=False,
    limit_choices_to={"enabled" : True})
    no_of_members = models.IntegerField()
```

Code for forms.py

```
from django import forms
from .models import Member
from .models import Grade
from .models import Section

class MemberUpdateForm(forms.ModelForm):
    name = forms.CharField(max_length=45, disabled=True)
    uid = forms.IntegerField(disabled=True)

class ActivityUpdateForm(forms.ModelForm):
    name = forms.CharField(max_length=45, disabled=True)
    id = forms.IntegerField(disabled=True)

class EventUpdateForm(forms.ModelForm):
    name = forms.CharField(max_length=45, disabled=True)
    id = forms.IntegerField(disabled=True)

class Meta:
    model = Member
    fields = '__all__'
```

MySQL Tables

1. Grade:

Primary key - id

Not null - id, name, description

desc grade;

	Field	Type	Null	Key	Default	Extra
▶	id	int	NO	PRI	NULL	auto_increment
	name	varchar(45)	NO		NULL	
	description	varchar(45)	NO		NULL	

select * from grade;

	id	name	description
▶	1	9	9th grade
	2	10	10th grade
	3	12	12
*	4	11	11
	NULL	NULL	NULL

2. School:

Primary key - id

Not null - id, name

desc school;

	Field	Type	Null	Key	Default	Extra
▶	id	int	NO	PRI	NULL	auto_increment
	name	varchar(45)	NO		NULL	
	description	varchar(45)	YES		NULL	

select * from school;

	id	name	description
▶	1	CHIREC	Chirec International School
*	NULL	NULL	NULL

3. Curriculum:

Primary key - id

Not null - id, name

desc curriculum;

	Field	Type	Null	Key	Default	Extra
▶	id	int	NO	PRI	NULL	auto_increment
	name	varchar(45)	NO		NULL	
	description	varchar(45)	YES		NULL	

select * from curriculum;

	id	name	description
▶	1	CBSE	CBSE
	2	IB	IB
*	NULL	NULL	NULL

4. Role:

Primary key - id

Not null - id, name

desc role;

	Field	Type	Null	Key	Default	Extra
▶	id	int	NO	PRI	HULL	auto_increment
	name	varchar(45)	NO		HULL	
*	description	varchar(45)	YES		HULL	

select * from role;

	id	name	description
▶	1	President	President
	2	Vice President	Vice President
	3	Treasurer	Treasurer
*	4	Secretary	Secretary
*	NULL	NULL	NULL

5. Year:

Primary key - id

Not null - id, name, enabled

desc year;

	Field	Type	Null	Key	Default	Extra
▶	id	int	NO	PRI	HULL	auto_increment
	name	varchar(45)	NO		HULL	
	description	varchar(45)	YES		HULL	
*	enabled	tinyint	NO		HULL	

select * from year;

	id	name	description	enabled
▶	1	2022	2022	0
	2	2023	2023	1
*	NULL	NULL	NULL	NULL

6. Type:

Primary key - id

Not null - id, name

desc type;

	Field	Type	Null	Key	Default	Extra
▶	id	int	NO	PRI	NULL	auto_increment
	name	varchar(45)	NO		NULL	
	description	varchar(45)	YES		NULL	
	fee	decimal(10,0)	YES		NULL	

select * from type;

	id	name	description	fee
▶	1	General Member	General member	550
	2	Board Member	Board member	0
*	HULL	HULL	HULL	HULL

7. Section:

Primary key - id

Not null - id, name, description

desc section;

	Field	Type	Null	Key	Default	Extra
▶	id	int	NO	PRI	NULL	auto_increment
	name	varchar(45)	NO		NULL	

select * from section;

	id	name	description
▶	1	A	Section A
	2	B	Section B
	3	C	C
	4	D	D
	5	E	E
	6	F	F
*	HULL	HULL	HULL

8. Member:

Primary key - id

Unique - uid + year_id

Foreign key - grade_id, section_id, curriculum_id, year_id, school_id, type_id, role_id

Not null - id, name, grade_id, section_id, uid, curriculum_id, year_id, school_id, type_id

desc member;

	Field	Type	Null	Key	Default	Extra
▶	id	int	NO	PRI	NULL	auto_increment
	name	varchar(45)	NO		NULL	
	grade_id	int	NO	MUL	NULL	
	section_id	int	NO	MUL	NULL	
	uid	int	NO	UNI	NULL	
	activity	tinyint	YES		NULL	
	curriculum_id	int	NO	MUL	NULL	
	year_id	int	NO	MUL	NULL	
	school_id	int	NO	MUL	NULL	
	type_id	int	NO	MUL	NULL	
	role_id	int	YES	MUL	NULL	
	payment	tinyint	YES		NULL	

Future Extensions

- ❖ In the future, we plan on integrating online payments for member registration using UPI using a third party. Students can then pay through the UPI system instead of paying in cash, reducing the manual work of taking a count of the funds coming in.

- ❖ We also plan on implementing a platform for people to communicate with us about any concerns and ideas through a feedback form. After their activity has been completed, they can let us know through this feedback form on any improvements.

Bibliography

- Grade 12 CBSE IP textbook
- Grade 11 CBSE IP textbook
- <https://getbootstrap.com/>
- <https://www.w3schools.com/>
- <https://stackoverflow.com/>