# "EVENT TRACKER"

A PROJECT REPORT SUBMITTED TO

## THE NATIONAL INSTITUTE OF ENGINEERING, MYSURU

(An Autonomous Institute under VTU, Belagavi)



In partial fulfilment of the requirements for Project work (Database Laboratory), fifth semester

## Bachelor of Engineering

### in

## Computer Science and Engineering

*Submitted by*

## Anvita Murthy (4NI17CS126)

## R K Chitra (4NI17CS062)

## Shashank H (4NI17CS119)

Under the Guidance of

Dr. Yuvaraju B.N
Professor,
Dept. of Computer
Science and Engineering

Mr. Narender M
Assistant Professor,
Dept. of Computer
Science and Engineering

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

## THE NATIONAL INSTITUTE OF ENGINEERING

Mysore-570 008

2019-2020

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
# THE NATIONAL INSTITUTE OF ENGINEERING



## *CERTIFICATE*

This is to certify that the project work entitled "**Event Tracker**" is a work carried out by **Anvita Murthy (4NI17CS126), R K Chitra (4NI17CS062)** and **Shashank H (4NI17CS119)** in partial fulfilment for the project work (Database Laboratory), fifth semester, Computer Science & Engineering, The National Institute of Engineering (Autonomous Institution under Visvesvaraya Technological University, Belagavi) during the academic year 2019-2020. It is certified that all corrections and suggestions indicated for the Internal Assessment have been incorporated in the report deposited in the department library. The project work report has been approved in partial fulfilment as per academic regulations of The National Institute of Engineering, Mysuru.

**Internal Guides**                                             **Professor & Head**

_____         _____               _____

Dr.Yuvaraju B.N            Mr.Narender M               Dr. V K Annapurna
Professor            Assistant Professor         Professor and Head
Dept. of CS&E           Dept. of CS&E              Dept. of CS&E
NIE, Mysuru             NIE, Mysuru               NIE, Mysuru

**Examiners**

_____                                       _____

Name:                                                Name:
Designation:                                         Designation:

# ACKNOWLEDGEMENTS

I would like to take this opportunity to express out profound gratitude to all those people who were directly or indirectly involved in the completion of this project. I thank each and every one who encouraged me in every possible way.

I would like to thank our Principal Dr G. Ravi for letting me be a part of this prestigious institution and letting me explore my abilities to the fullest.

I would like to extend my sincere gratitude to Dr V K Annapurna, our H.O.D for being the source of inspiration and instilling an enthusiastic spirit in me throughout the process of project making I would like to express heartfelt gratitude towards my project guides Dr Yuvaraju B.N and Mr Narender M and for their constant guidance, valuable knowledge and experience.

I am thankful to my parents, family and friends for their constant support, inspiration and encouragement for the helping hand in all those who supported me directly or indirectly in my academic process.

Thank you.

**- Anvita Murthy**

**- R K Chitra**

**- Shashank H**

# Table of Contents

# List of Figures

**Chapter 1**

# INTRODUCTION

NIE is a college that conducts several events throughout the year. These events are conducted by various clubs and some by the college itself. Keeping track of the innumerable events conducted by each of the clubs and the additional ones conducted by the college itself is a cumbersome task. This is the problem at hand that needs to be solved.

This project aims at providing a database management system for all the clubs in NIE. It is a platform to provide the clubs of NIE with a more efficient means of managing the event notifications in a such a way that it reaches the maximum number of students in a user-friendly way.

Our project 'Event Tracker' provides a solution to the students where they can easily look up into our website for all the updates and information that they need about the ongoing and upcoming events happening in our college. Having a consolidated website for this makes the task simpler.

Our project provides a basic interface for the users in the form of a website to view all the clubs in NIE and details about each club and their events that will be organized along with many other features to aid the students.

The students, each of them will have an account of his/her which they can login using their credentials that will redirect them to our main website where the content that they need will be displayed.

The website interface is made using HTML, CSS and JavaScript. Databases are maintained using MySQL and the website is hosted using Flask module for Python. Any fairly modern computer with an operating system that supports the web server can be used for running the website.

**Chapter 2**

# SYSTEM ANALYSIS

System analysis is a detailed study of the various operations performed by a system and their relationships within and outside of the system. Here the key question is - What all problems exist in the present system? What must be done to solve the problem?

Analysis begins when a user or manager begins a study of the program using existing system. During analysis, data collected on various files, decision points and transactions handled by the present system. The commonly used tools in the system are Data Flow-Diagrams, interviews, etc. training experience and common sense are required for collection of relevant information needed to develop the system. The success of the system depends largely on how clearly the problem is defined, thoroughly investigated and properly carried out through the choice of the solution.

A good analysis model should provide not only the mechanism of the problem understanding but also the framework of the solution. The proposed system should be analyzed thoroughly in accordance with the needs.

## 2.1 Existing and Support System

The current system involves notifying students of the upcoming events through class announcements or by circulating messages on each class' WhatsApp group. Information about the events conducted by the institution is sent via email. More often than not, the notifications sent through WhatsApp or email go unnoticed. Many students are unaware about the events being conducted around them and miss their opportunities to take part, hindering their growth.

## 2.2 Proposed System

The main objective of this mini project is to let students of our college know about the events that are happening in the college and to provide a platform for the Event-handlers to notify students about their events so that students never miss out on an opportunity to attend an event that they might be interested in.

Students need to create an account in order to be able to use our services. After entering their details and signing up, they can view all the events posted and register for the same. All event-handlers get a ready-to-use account which allows them to post events. While posting events, event-handlers need to enter details about the event like the venue, number of seats available and time.

The event registration page displays a real time count of the remaining seats. Students interested in events conducted by specific club can filter out other events by using the filter option. If students decide not to attend an event after registering, they can do so by clicking 'Unregister' and the seat is made available for registration again.

All events which go on for more than a day are termed 'Big-event' and will be displayed above all other events in the event registration page.

Our project uses Python 3.6 for the backend. The Flask library provides a hosting service for http and ftp responses. MySQL is used for the database. It is an RDBMS which can be easily integrated with Flask python. These two systems handle the server side. The frontend of our website uses HTML, CSS and JS components are used to stylize the website.

The Flask Project can be run on a server with network access. This is important because the website grabs font, CSS and icon files from Google's server. The website requires a fairly modern web browser with JS enabled.

## Software Details:

| Program or package | Version used |
| --- | --- |
| Python | 3.6.7 |
| MySQL | 8.0.17 |
| Flask | 1.1.1 |
| HTML | 5 |
| CSS | 3 |

## Software and Hardware Requirements:

**Server side:**

Hardware:

1: A dual core CPU @ 2.5GHz

2: minimum of 256 MB ram

3: 20 GB of storage

4: Fast and high bandwidth internet connection

Software:

1: Compatible version of any Linux or Windows OS

2: Python 3.6.7

3: Flask

**User side:**

Hardware:

1: A dual core CPU @ 2.5GHz

2: minimum of 1 GB ram

3: 20 GB of storage

4: Internet connection

5: Keyboard and mouse setup

Software:

1: Compatible version of any Linux or Windows OS

2: A web browser with JS enabled.

**Chapter 3**

# SYSTEM DESIGN

## 3.1 Design

The website is made using Flask on Python 3.6.7. MySQL is used as the database manager. The frontend is made using HTML, CSS and JS. The details of these libraries are given below:

## 3.1.1 HTML

HTML or HYPERTEXT MARKUP LANGUAGE is a standard markup language used to create webpages. The main function of a web browser is to read HTML documents and compose them into visible or audible web pages. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms, may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by *tags*, written using angle brackets. Tags such as <img/> and <input/> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. The browser does not display the HTML tags that are used to create the webpages, but uses the tags to interpret and represent the content of the page. HTML describes the structure of a website semantically along with cues for presentation, mailing it a markup language rather than a programming language.

## 3.1.2 Cascading style sheets

Cascading style sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interface written in HTML and Hotplate language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to reading in speech or on other media. Web browsers can refer to CSS to define the look and layout of text and other material. The W3C maintainer of both the HTML and the CSS standards, encourages the use of CSS over explicit presentational HTML. Along with HTML and JavaScript, CSS is a technology used by most websites to create visually engaging web pages, user interface for web applications, and user interface for many mobile applications. CSS is designed primarily to the separation of document content from document presentation, including aspects such as the layout, colors and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .CSS file, and reduce complexity and repetition in the structural content.

### 3.1.3 JavaScript

JavaScript, often abbreviated as JS, is a high-level, dynamic, prototype-based, multi-paradigm, and interpreted programming language. Alongside HTML and CSS, JavaScript is one of the three core technologies of World Wide Web Content production. It is used to make webpages interactive and provide online programs, including video games. The majority of websites employ it, and all modern web programs support it without the need for plug-ins by means of a built-in Java script engine.

### 3.1.4 MySQL

MySQL is an open source relational database management system (RDBMS). MySQL enables data to be stored and accessed across multiple storage engines, including InnoDB, CSV, and NDB. MySQL is also capable of replicating data and partitioning tables for better performance and durability. MySQL users aren't required to learn new commands; they can access their data using standard SQL commands.

For security, MySQL uses an access privilege and encrypted password system that enables host-based verification. MySQL clients can connect to MySQL Server using several protocols, including TCP/IP sockets on any platform. MySQL also supports a number of client and utility programs, command-line programs and administration tools such as MySQL Workbench. Different programming languages have modules and libraries that enable the use of MySQL.

### 3.1.5 FLASK

Flask is a micro web framework written in Python. It is classified as a micro-framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Extensions are updated far more regularly than the core Flask program. Flask is commonly used with MongoDB, which gives it more control over databases and history.

## 3.2 TABLES USED AND ER DIAGRAMS

### 3.2.1 Login Table

| Field | Type | Null | Key | Default | Extras |
|-------|------|------|-----|---------|--------|
| name | Varchar | Yes | | NULL | |
| email | Varchar | No | PRIMARY | NULL | |
| USN | Varchar | Yes | | NULL | |
| pno | Varchar | Yes | | NULL | |
| passwd | Varchar | Yes | | NULL | |
| Eh | Varchar | Yes | | N | |

The Login table contains all the details entered during the signup process. The table contains information of both, the event-handlers and the students. The details of event-handlers are added manually by the creators of the website. Email is the unique identifier of each student/event-handler and is made the primary key of the table. The 'eh' column tells if an account is a student account or an event-handler account. It is set to 'No' by default.

### 3.2.2 Events Table

| Field | Type | Null | Key | Default | Extras |
|-------|------|------|-----|---------|--------|
| cname | Varchar | Yes | | NULL | |
| ename | Varchar | No | PRIMARY | NULL | |
| edescp | Varchar | Yes | | NULL | |
| seats | Int | Yes | | NULL | |
| dt | Varchar | Yes | | NULL | |
| tm | Varchar | Yes | | NULL | |
| venue | Varchar | Yes | | NULL | |
| big event | Varchar | Yes | | NULL | |

The Events table contains details of the events posted. It includes the club name, event name, event description, date, time and venue. The 'big event' field tells if an event is a big event or not. The ename is the unique identifier and is made the primary key of the table.

### 3.2.3 Notify Table

| Field | Type | Null | Key | Default | Extras |
|-------|------|------|-----|---------|--------|
| email | Varchar | Yes | Foreign | NULL | |
| cname | Varchar | Yes | | NULL | |

If a student subscribes to a club, his email and the club name automatically get added to this table. Whenever the club a student is subscribed to, posts an event, an email notification is sent to the student. If a student unsubscribes from a club, his/her details are automatically deleted from the table. Here, the email is a foreign key which references the Login table.

### 3.2.4 Pevents Table

| Field | Type | Null | Key | Default | Extras |
|-------|------|------|-----|---------|--------|
| cname | Varchar | Yes | | NULL | |
| ename | Varchar | Yes | | NULL | |
| edescp | Varchar | Yes | | NULL | |
| dt | Varchar | Yes | | NULL | |
| big_event | Varchar | Yes | | NULL | |

The Pevents table contains details of the events a student attended in the past. The date (dt) field of the Events table is compared of the system date. If the system date has gone past the event date, the above fields of that particular record is deleted from the events table and added to the Pevents table.

### 3.2.5 Reg Table

| Field | Type | Null | Key | Default | Extras |
|-------|------|------|-----|---------|--------|
| ename | Varchar | Yes | | NULL | |
| sname | Varchar | Yes | | NULL | |
| Usn | Varchar | Yes | | NULL | |
| email | Varchar | Yes | Foreign | NULL | |
| ph | Varchar | Yes | | NULL | |
| memb | Varchar | Yes | | NULL | |

When a student registers for an event, his details are automatically added to this table. Whenever a student visits the 'My Events' page, the 'Upcoming' and the 'Earlier' events that he/she had registered for, are extracted using a join on this table and the Events table/Pevents table respectively.
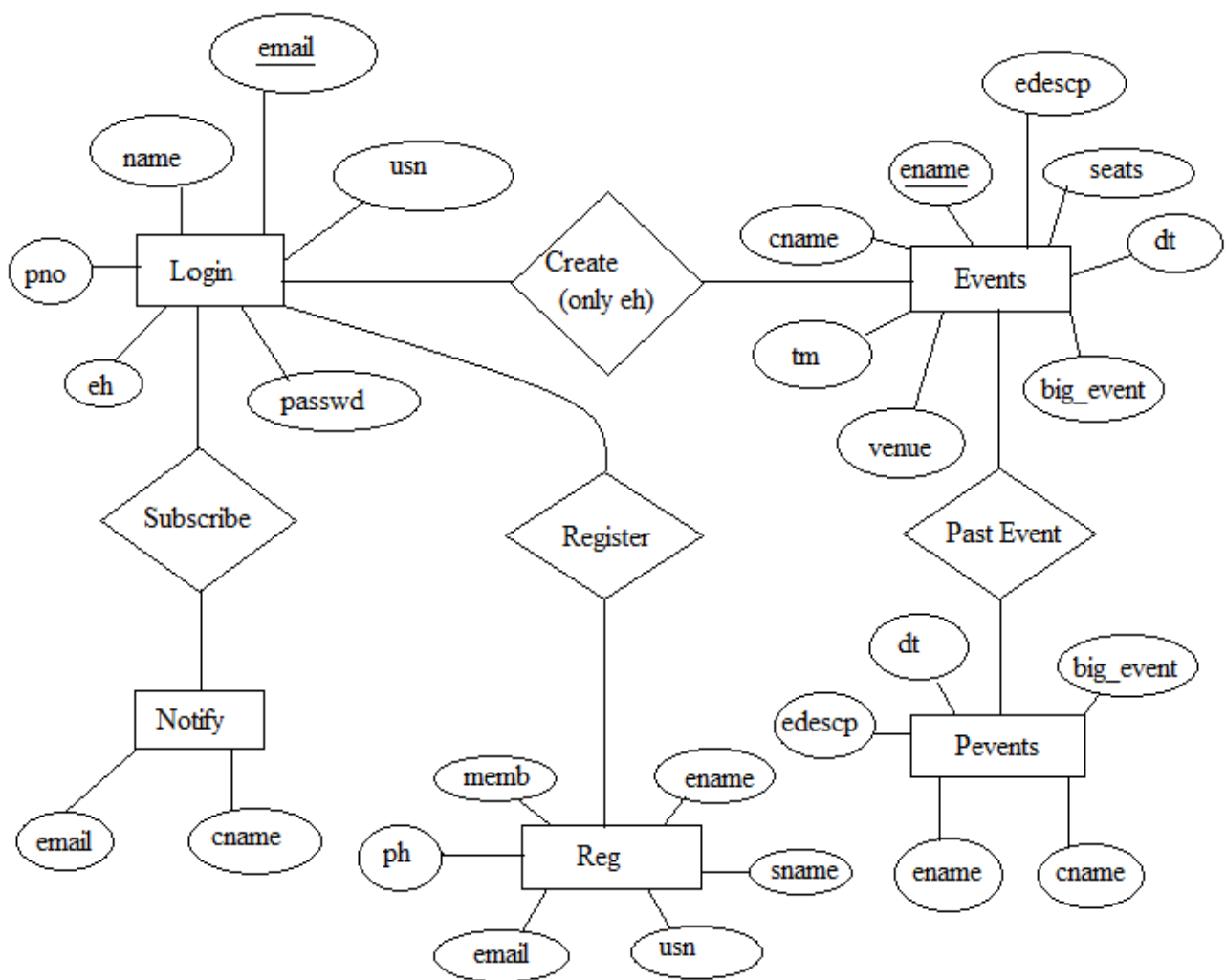
**Entity Relationship Diagram for Event Tracker**

**Chapter 4**

# SYSTEM IMPLEMENTATION

The first step in using the website is logging in and on opening the website the sign-up page is displayed where a user enters various fields as per the requirements. Based on this, the actions that can place are divided into sections:

- A user opens our website:

He/she is supposed to enter his/her name, USN, contact, password and E-mail which is the primary key. The user will be asked to re-enter their password to confirm there is no mistakes. After entering the following credentials, they can successfully create an account in our website. There is an option where, if they've already created an account they can click on 'login' button and it redirects them to the login page. The login page requires that the user enter his E-mail and password to go to the home page of the website.

- A user enters the home page:

The home page is from where you are redirected to different page to explore the various functionalities incorporated in our website. The user can navigate to the various clubs to get information about upcoming events, previous events and brief description about the club. A dropdown menu called 'events' gives an option to the user to choose between 'Upcoming Events' or 'Previous Events'. The upcoming events option is where it redirects you to the page where the list of events is displayed and you can register to the events of your choice.

- Registering for events:

This page lets to register for upcoming events. Filtering the events based on the clubs can also be done. All the upcoming events irrespective of the clubs are displayed and filtering can be done as per user's wish.

- User Profile:

A dropdown menu called 'My Profile' lets you to navigate between 'My Events' and 'Settings'. The first option redirects you to the page where an account of the events that you've previously attended will be displayed as well as the upcoming events that you've registered and the second option redirects to the page where you are allowed to edit your details like name, USN and also lets you to change your password.

The several components of Database Management System that we've implemented in this website are as follows:

- Trigger:

The trigger 'update_seat_count' decrements the number of seats whenever someone registers for an event since there are limited seats allotted for an event.

- Procedure:

Whenever someone tries to view the upcoming events, a procedure 'update_events' is called. It inserts events whose date is lesser than current system date, from the events table to Pevents (previous events) and deletes those entries from the events table after insertion. Also, whenever a user unregisters from an event a procedure 'unregister' is called. It deletes the entry of the user for that particular event and updates the event table by incrementing the number of seats by one.

- View:

There's a view called 'details' that stores all the details of the user who has logged in; hiding the password field. These details are used in the 'Edit my details' page and 'Register' page. These details are extracted from the database and displayed on the pages.

- Index:

The E-mail field is made as an index in the register table. Since the same user can register for multiple events and this table has the registrations not only for upcoming events, but for earlier events also and since there'll be so many users, an index is used on the E-mail column to get the query results faster.

- Foreign Key:

Since E-mail is a primary key in the login table, any other table that has 'E-mail', is a foreign key reference to this one and it has this ON UPDATE CASCADE option with it, which means whenever the primary key is updated in the main table, the foreign key is also updated in the other tables (when the students choose to edit their details).

- Function:

The function 'event_num' takes the E-mail ID as the parameter and the returns the number of upcoming events a particular student has registered to irrespective of the clubs.

**Chapter 5**

# SYSTEM TESTING

## 5.1 Test Cases

System testing is performed to evaluate the compliance of the system with the corresponding requirements. In this case, system testing is performed on the database, which involves checking tables, results of queries, triggers, procedures etc.

The various test cases involved are:

1. Student Registration
2. Logging in of users/event handlers.
3. Editing details
4. Changing Password
5. Entering the URL of the pages, when not logged in.
6. Entering the URL of 'Create An Event' page when logged in, as a regular user (not event handler)
7. Attempting to re-register for an event
8. Attempting to register for an event where no more seats are available
9. Unregister from event

## 5.2 Working

**1. Student Registration**: When a student attempts to sign up and enters the credentials required for the same, if the email or the USN entered by him already exists in the database, an error message is displayed and account creation is unsuccessful. Also, the user is asked to re-confirm the password. If the passwords don't match, an error message is displayed and once again account creation is unsuccessful. If the details entered don't already exist in the database and both the passwords match, the account is created.

**2. Logging in of students/event handlers**: When a login attempt is made, the credentials entered are verified and the login is successful only if the credentials match the ones in the database. If not, an error message is displayed. Students and event handlers both are directed to a very similar home page. Event handlers however have an added feature of 'Create' to announce upcoming events of their clubs.

**3. Editing Details**: Students have the flexibility of editing their details, which are required for the registration form while registering for events. Once again, while editing details, if the new USN or email are found to exist in the database already, the changes aren't made and instead, an error message is

displayed. However, if the new values entered don't already exist in the database, the updates are saved and reflected in the database.

**4. Changing Password**: While changing passwords, the user is required to enter the current password, new password and re-enter the new password. If (i) Current password entered is incorrect or (ii) the new password and re-entered password fields don't match or (iii) new password is the same as the current password, an error message is displayed. If none of the above is true, then the password is changed and updated in the database as well.

**5. Entering URL of pages when not logged in**: When the user hasn't logged in and attempts to enter the URL of a page, they're redirected to the login page. This is achieved with the help of sessions in flask, which is the time interval when a client logs into a server and logs out of it. The data, which is needed to be held across this session, is stored in a temporary directory on the server. Session used in this case is a dictionary which has 'email' and 'eh'(Event handler) as its keys and the corresponding email id and either 'yes' or 'no' as its values. Whenever a user login, these values are initialized. Whenever a user logs out, these values are popped from the dictionary. Therefore, a user cannot access the pages of the site unless they've logged in.

**6. Entering the URL of 'Create' page when logged in, as a regular user (not event handler)**: If a user who is logged in tries to access the 'Create' page by entering the URL and the user is not an event handler, the user is redirected to the home page instead. 'Create' is a feature that is available only to event handlers and not regular users. This is implemented by using sessions in flask. One of the keys in the dictionaries is eh (stands for event handler), which is initialized when the user logs in. The user is granted access to this page only if they're event handlers. If not, they're redirected to the home page.

**7. Attempting to re-register for an event**: If a user attempts to register for an already registered event, the user is redirected to the 'My profile' page that lists all the events the user has registered for, along with a flash message saying 'You've already registered for this event.' This is accomplished by running a query to check if there are entries in the registration table with that email for that specific event. If a non-empty set is returned, it is understood that the user has already registered and no entry is made in the table. If an empty set is returned, the user is registered for the event (an entry is made in the table) and the user is redirected to the 'My Profile' page along with a flash message confirming the registration.

**8. Attempting to register for an event where no more seats are available:** Each time a user registers for an event, the number of seats available is decremented. When the last seat gets taken and the number of seats is 0, the registration link is disabled and the column value changes to 'N/A' instead.

**9. Unregister from an event:** When a student chooses to unregister from an event, the corresponding entry is removed from the database and that seat becomes vacant again i.e. the number of seats is incremented by 1 and if it was previously 0 (due to which the registration link was disabled), the registration link is enabled once again. If the user tries to re-register after unregistering, it doesn't display an error message and allows the student register again, if required.

# Chapter 6

# RESULTS



**Fig 6.1- Sign up Page**

This is the page used for account creation when a student initially wants to sign up. Once the account is created, the student can login.



**Fig 6.2 – Login Page**

After account creation, the user can enter their details and login. The login credentials entered are verified and if they match the ones in the database, login is successful. If not, an error message is displayed.
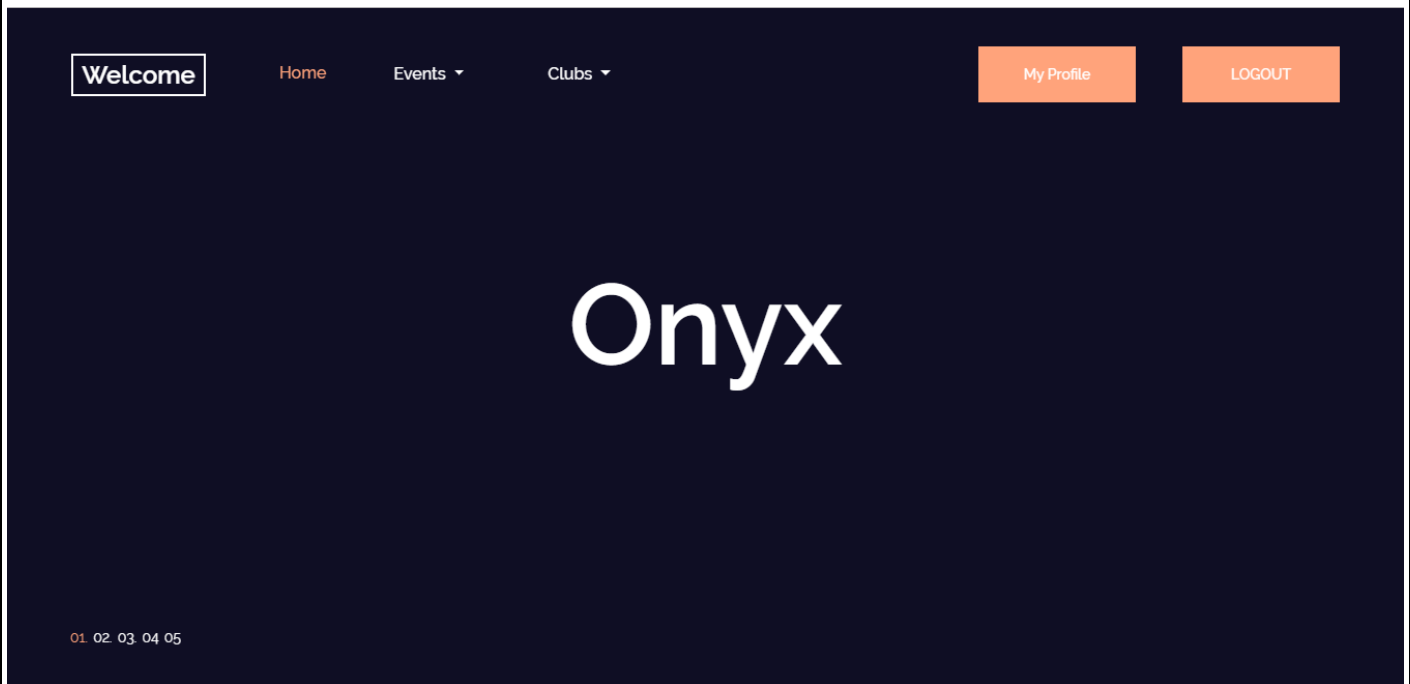
**Fig 6.3- Home page**

After logging in, users are directed to the home page that displays the clubs on a slider, with a link to the corresponding club page. The navigation bar has a dropdown titled 'events', which has links to the upcoming events page and the earlier events page (List of events that took place earlier that year and their details.)
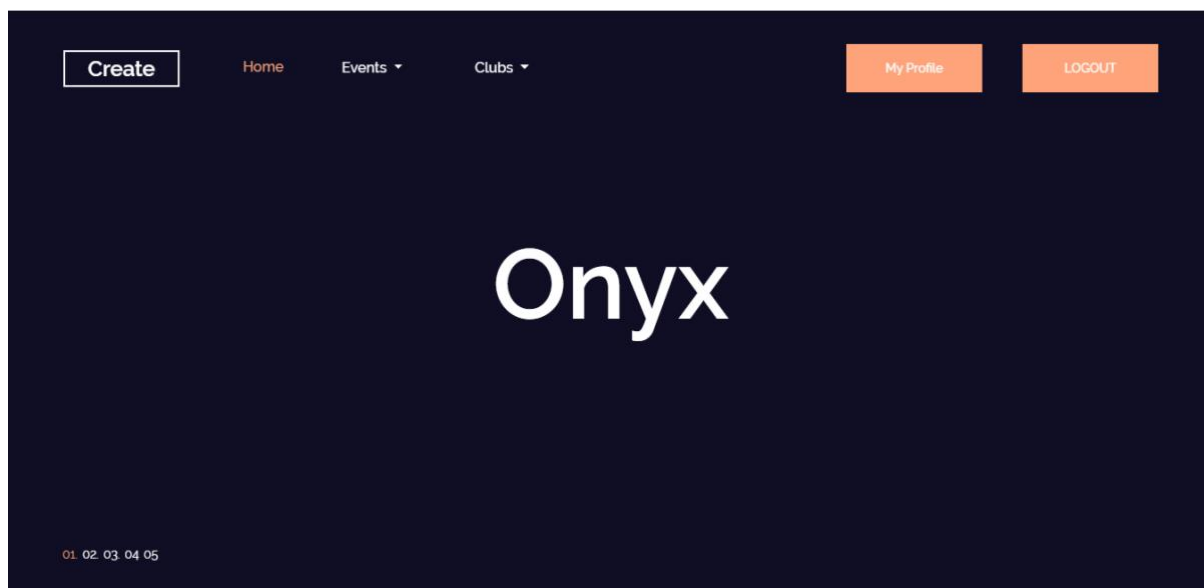


**Fig 6.4 – Home page for event handlers**

The home page for event handlers is very similar to that of regular users, except that it has an extra option called 'Create' that can be used to create announcements about upcoming events, a feature available only to verified event handler accounts.

**Fig 6.5- Upcoming events**

This page displays all the upcoming events dynamically. Whenever an event is created by an event handler, it is automatically reflected here. This table lists major events first and then the regular events. Details about each event are displayed and the number of seats is decremented whenever a student registers, and the registration link is disabled once the number of seats reaches zero.
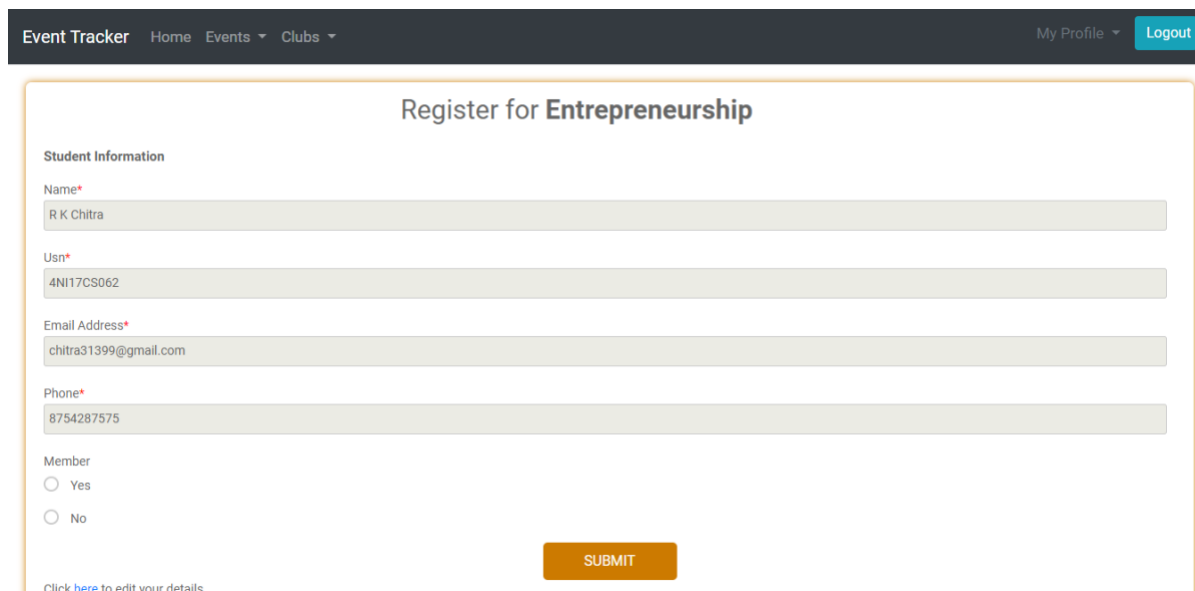


**Fig 6.6 – Earlier events with filter**
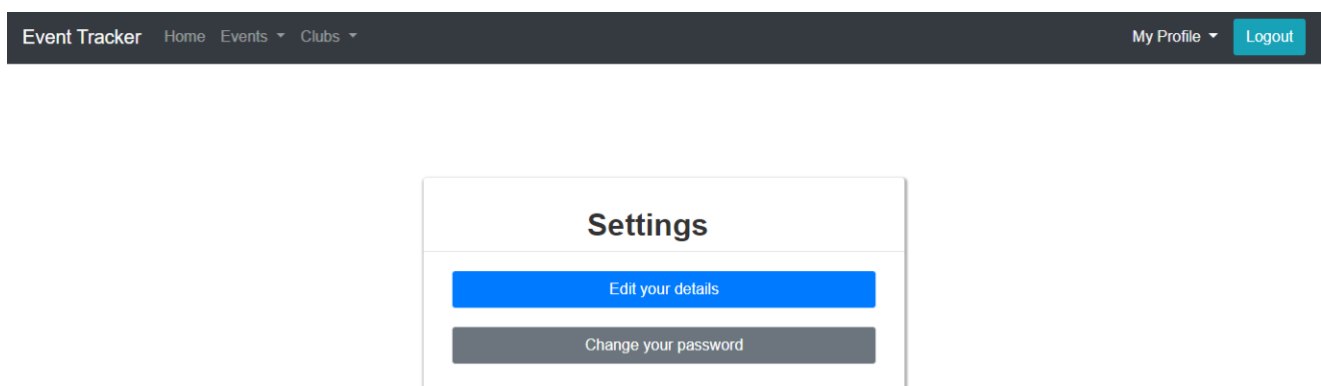
This page the events conducted earlier that year. The 'filter events' is used to filter the events displayed, based on the club. This feature is available in the 'Upcoming events' page as well and functions in a similar manner.



**Fig 6.7- Registration Form**

This is an auto-filled page to register for any upcoming events. The text inputs in this page are disabled and can be edited by only editing the details of the user themselves (by clicking the link provided) and any changes made will be reflected on this page.



**Fig 6.8 – Settings Page**

The settings page can be accessed through the dropdown titled 'My Profile' that has the dropdown items 'My Events' and 'Settings' respectively. This contains links to the 'Edit your details' and 'Change your password' pages.

**Fig 6.9 – Edit your details Page**

This page displays the user's details and lets the user edit their details, provided the changes made in fields like email and USN aren't already present in the database. The 'Save' button submits the form whilst the 'Cancel' button redirects the user back to the settings page, without saving any changes (if any) made.



**Fig 6.10 – Change your password page**

This page allows the user to change their password provided they enter the 'Current password' correctly and the 'New password' and 'Confirm Password' fields must match and shouldn't be the same as the current password. 'Cancel' redirects the user back to the settings page whilst the 'Save' button submits the form.



**Fig 6.11 – Student Participation**

This is the 'My Events' page which can be accessed using the 'My Profile' dropdown menu. This lists all the upcoming events the user has registered for, and the earlier events the student has registered for in the past are displayed in another table.



**Fig 6.12 – Unregister from event**

Students can choose to unregister from an event. Once unregistered, they can always re-register, if seats for the event are still available. The seat becomes vacant once the user unregisters and the entry is removed from the table.



**Fig 6.13 – Club Page**

Users can 'Subscribe/Unsubscribe' on club pages. The 'Subscribe' option registers their email and they receive notifications whenever the club(s) they've subscribed to posts an event. The 'Unsubscribe' option removes their email from the respective club's group of subscribers and they no longer receive emails on unsubscribing.
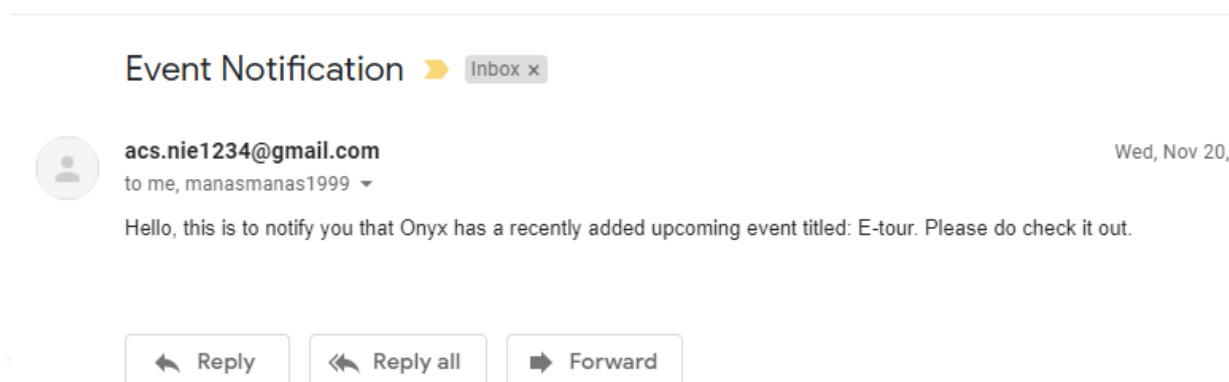


**Fig 6.14 – Notifications via email**

Users receive auto-generated emails whenever the club(s) they've subscribed to announces an upcoming event. They cease receiving emails once they unsubscribe.



**Fig 6.15 – Create an event**

This page is used to create an event which requires the event handler to enter some details like the name of the event, event description, number of seats allocated, date when the event is scheduled to take place, time, venue and if it is a major event or not (Major events are those events that go on for more 1 day).

# CONCLUSION AND FUTURE ENHANCEMENTS

The main purpose of this site is to act as an interface between event handlers and students, by enabling event handlers to announce upcoming events and this platform lets students register for events they're interested in. It also allows students to view all the events they've registered for so far, and allows them to unregister if they won't be able to make it, thereby making that seat open for registration again. Students also have the option of subscribing to clubs of their choice to be notified via email whenever the club announces an event.

A good future enhancement would be allowing event handlers to add updates about the events and all the people who have registered receive an email whenever the event handler updates the event. Another future enhancement would be to include an online payment option for the events that have a registration fee. These enhancements couldn't be implemented due to time constraints and since online payment transactions involve cyber-security and vulnerable to attacks by malicious entities.

# REFERENCES

- The Database book: Principles and Practice using MySQL, Narain Gehani, Universities press (India) private Limited 2008
- Fundamentals of Database System, Elmasri and Navathe, Addison-Wesley, 5th edition, 2007
- www.w3school.com  for HTML, CSS, JavaScript.
- https://getbootstrap.com/ for front end.
- https://fonts.google.com/ to select fonts for the front-end pages.
- www.tutorialspoint.com for Flask tutorials.
- https://www.youtube.com/ for Flask tutorials.
- http://stackoverflow.com/ for solving our errors.