Visvesvaraya Technological University, Belagavi – 590018

PROJECT REPORT
ON

End to End Event Management System

Submitted in partial fulfillment of the requirements for the degree

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE & ENGINEERING

Submitted by

Nireekshith	4SO18CS081
Novin Misquith	4SO18CS086
Pranith Rao	4SO18CS088
Vaibhav K Suvarna	4SO18CS120

Under the guidance of

Ms. Gayana M N
Assistant Professor, Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ST JOSEPH ENGINEERING COLLEGE
(Affiliated to VTU Belagavi, Recognized by AICTE, Accredited by NBA)

Vamanjoor, Mangaluru-575028, Karnataka

2021-2022

ST JOSEPH ENGINEERING COLLEGE

(*Affiliated to VTU-Belagavi, Recognized by AICTE, B. E. (CSE) - NBA and NAAC Accredited*)

Vamanjoor, Mangaluru-575028

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work entitled "**End to End Event Management System**" carried out by

Nireekshith	**4SO18CS081**
Novin Misquith	**4SO18CS086**
Pranith Rao	**4SO18CS088**
Vaibhav K Suvarna	**4SO18CS120**

bonafide students of VIII semester (Computer Science & Engineering) in partial fulfillment for the award of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belagavi during the year 2021-2022. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

Ms. Gayana M N	**Dr Sridevi Saralaya**	**Dr Rio D'Souza**
Project Guide	**HOD - CSE**	**Principal**

External Viva

Name of the Examiner **Signature with date**

1.

2.

ACKNOWLEDGEMENT

We dedicate this page to acknowledge and thank those who helped shape this project. Without their guidance and help, the experience while constructing the dissertation would not have been so smooth and efficient.

We sincerely thank **Ms. Gayana M N, Assistant Professor**, Computer Science and Engineering, for her guidance and valuable suggestions. We extend our heartfelt gratitude to **Ms. Sunitha Guruprasad** and **Dr Harivinod N, Associate Professor**, Computer Science and Engineering for being the project coordinators and giving constant supervision as well as providing us the necessary information regarding the project.

We owe our profound gratitude to **Dr Sridevi Saralaya, Head of the Department**, Computer Science and Engineering, whose kind consent and guidance helped us to complete this work successfully.

We are extremely thankful to Principal **Dr Rio D'Souza**, Director **Rev.Fr. Wilfred Prakash D'Souza** and our Assistant Directors **Rev.Fr. Rohith D'Costa** and **Rev.Fr. Alwyn Richard D'Souza** for their support and encouragement.

We would like to thank all our Computer Science and Engineering staff members who have always been with us extending their support, precious suggestions, guidance and encouragement through the project.

We also like to extend thanks to our friends and family members for their continuous support.

ABSTRACT

Any event must be carefully managed and coordinated between numerous organizers in order to be successful. In written manual records, a major issue is a lack of communication and the lack of updated participant records. Managing many events at the same time will simply add to the difficulty. Our project intends to simplify this by incorporating cutting-edge computer science technology.

End-to-End event management system is the management of events which consists of the creation, development and updating of small and/or large-scale personal or corporate events. Our project explains how to use a web-based interface to manage an event. The primary goal of creating an event management interface is to provide the institution with a single application that will aid in the organization and management of all events. The outcome of this project is a simple and unified interface for managing and monitoring event progress on the move, to smoothly perform the entire event, and to keep all events digitized.

The interface of our website will provide options for a relatively easy data input text-boxes that will be properly labeled. It will also have a user-friendly view of the whole system with simple and easy undertaking of action-driven processes as command buttons are functionally labeled. With all these, target users of this website will relatively won't find it difficult to use it.

Our application makes it simple to organize an event and manage it, as well as track progress on the go, a completely seamless event on the go. Data from participants and events are better visualized on the dashboard. All Integrated judgement within the web app has an automated certificate. Overall, this application minimizes the amount of extra work that event organizers have to do when administering a large-scale event.

TABLE OF CONTENTS

Page Number

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1 Background

The "Event Management System" has been developed to override the problems prevailing in the practicing manual system. This application is supported to eliminate and, in some cases, reduce the hardships faced by this existing system. Moreover, this system is designed for the particular need of the organizations to carry out operations in a smooth and effective manner.

Any event must be carefully managed and coordinated between numerous organizers in order to be successful. Every organization, whether big or small, has challenges to overcome and managing the information of Activity, Event, Attendees, Payment, Conductors. In written manual records, a major issue is a lack of communication and the lack of updated participant records. Managing many events at the same time will simply add to the difficulty. Our project intends to simplify this by incorporating computer science technology.

End-to-End event management system is the management of events which consists of the creation, development and updating of small and/or large-scale personal or corporate events. Our project explains how to use a web-based interface to manage an event. The primary goal of creating an event management interface is to provide the institution with a single application that will aid in the organization and management of all events. This also provides an automatic certificate of participation and winner certificates in one single app. The interface in turn helps in generating and mailing of certificates too. The outcome of this project is a simple and unified interface for managing and monitoring event progress on the move, to smoothly perform the entire event, and to keep all events digitized.

1.2 Problem statement

Any event to be successful has to be has to be properly managed and coordinated between the various organizers. Lack of communication and non-availability of updated records of participants is a huge problem in manual written records. Multiple events at a single time will only increase this complexity our project aims on simplifying this. Our project demonstrates on managing an event using a web-based interface. The main objective in developing an event management interface is providing the institution a single application which will help them to organize and manage all the events.

1.3 Scope and Importance

Our application makes it simple to organize an event and manage it, as well as track progress on the go, a completely seamless event on the go. Data from participants and events are better visualized on the dashboard. All Integrated judgement within the web app has an automated certificate. Overall, this application minimizes the amount of extra work that event organizers have to do when administering a large-scale event. Few events which can be conducted through mobile can be integrated into this application. Adding map location of each event can make it easy for participants to reach the event spot.

CHAPTER 2

SOFTWARE REQUIREMENT SPECIFICATION

CHAPTER 2

SOFTWARE REQUIREMENT SPECIFICATION

2.1 Functional requirements

- Integrated Judgement within the web app.
- Dashboard: Better visualization of the data of participants and events.
- Secure registration and profile management facilities for different users.
- Generates the alert via e-mail.
- Provide schedule/timetable without any of the clashes among judges, day, time and room that must be visible to all.
- Provides an automatic certificate of participation and winner certificates in one single app. The interface in turn helps in generating and mailing of certificates too.

2.2 User interfaces

The participant’s dashboard will include:

- A list of all the events the student can participate in.
- A registration form of each event.

The organizer’s dashboard will include:

- A form to register co-organizer’s events.
- A list of students participating, judges involved and all the other details of a particular event will be visible.

The co-organizer’s dashboard will include:

- List of participants of his particular event.
- Form to add/invite judges.

The judges’ dashboard will include:

- A form to enter marks for each participant.
- An option to submit the form once the judging is done.

2.3 Software Requirements

- **Operating system:** No particular OS is required because web application is platform independent
- **Database:** SQLite
- **Programming language:** Python, JavaScript.
- **Framework used:** Flask, Bootstrap, JQuery
- **Frontend tools:** HTML, CSS, JavaScript, Sweetalert.

2.4 Hardware Requirements

- **Processor:** INTEL®CORE™ i57200U
- **Speed:**2.50 GHz
- **RAM:** 4.00 GB
- **System Type:** 64-bit Operating System

CHAPTER 3
SYSTEM DESIGN

CHAPTER 3

SYSTEM DESIGN

3.1 Abstract Design

3.1.1 Architectural diagram

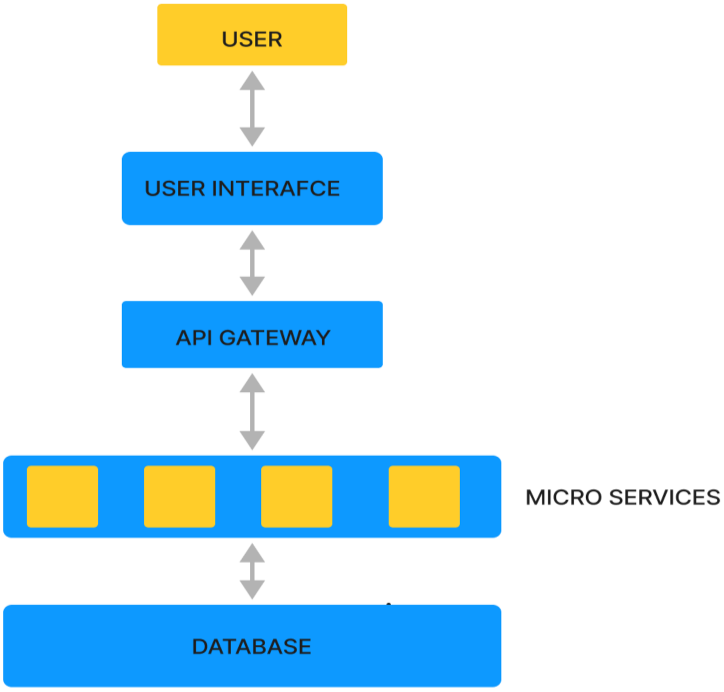


Fig. 3.1.1 Architectural diagram

End to End Event management is the management of events which consists of the creation, development and updating of small and/or large-scale personal or corporate events. Events are organized by almost all universities, but all work from registration to certification is done by hand and is not digitalized. Our project demonstrates on managing an event using a web-based interface. The main objective in developing an event management interface is providing the institution a single application which will help them to organize and manage all the events.

Organizer adds the co-organizers who add the events which are visible to the participants and they registered for the events of their interest and judges assigned to each events evaluate the participants' performance and give scores according to it which will help in issuing certificates to the participants. An architecture diagram is a graphical representation of a set of concepts, that are part of an architecture, including their principles, elements and components. The diagram explains about the system application in perception of overview of the system.

3.1.2 Use Case diagram

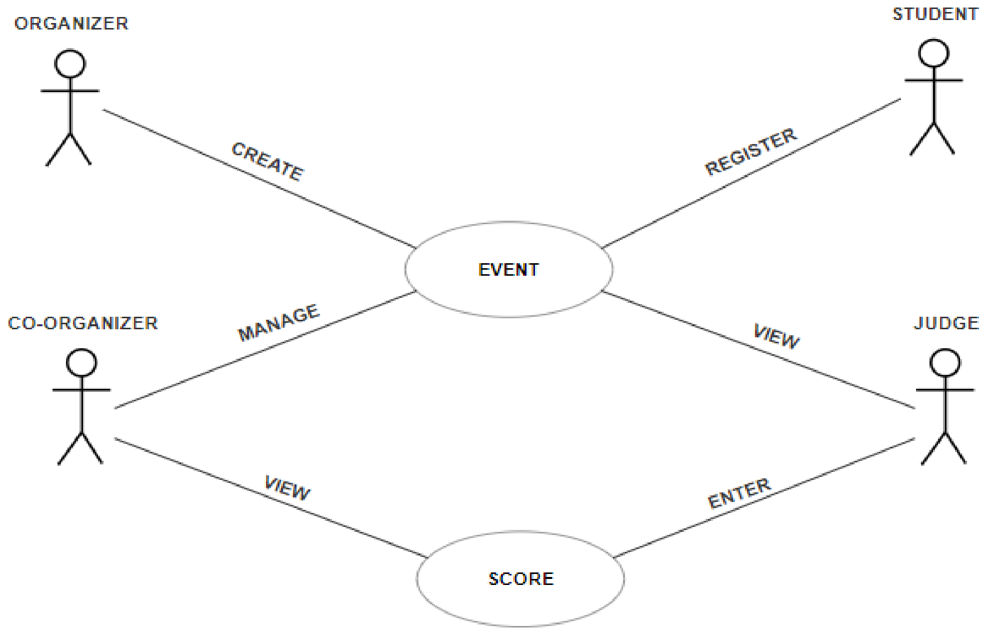


Fig. 3.1.2 Use Case diagram

Detailed Description of the Use cases:

- **Use Case**: Create Event

  Summary: The interface allows the organizer to create multiple events.

  Actor: Organizer

  Description: The organizer can create events involved in the program that can be participated by the participants and managed by the co-organizers.

- **Use Case**: Register Event

  Summary: The interface allows the participants to register and participate in the event.

  Actor: Student

  Description: The participants can register for the events of their interest. The number of events in which they can participate can be limited based on the restrictions.

- **Use Case**: Manage event and view scores

  Summary: The interface allows the co-organizer to manage event and view the scores submitted by the judges.

  Actor: Co-Organizer

  Description: The co-organizer manages the event he is assigned to and can also view scores that are submitted by the judges on basis of the performance of the participants in that particular event

- **Use Case**: View event and submit scores.

  Summary: The interface allows the judges to view the events and submit scores on basis of the performance of the participants.

  Actor: Judge

  Description: The judge can view the details of the event he is assigned to judge and also can submit the scores after judging the performance of the participants.

3.2 Functional Design

3.2.1 Data Flow diagram

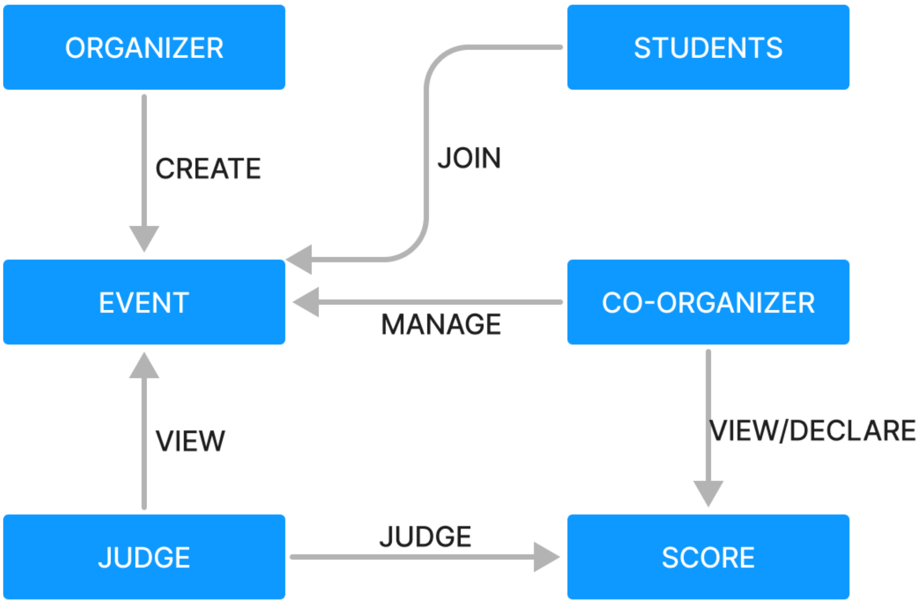


Fig. 3.2.1 Data Flow diagram

The dataflow diagram of this project consists of all the various aspects a normal flow diagram requires. This dataflow diagram shows how organizer adds events and assigns co-organizers to manage them, how students join the events of their interest and how judges judge the performance of each participant in various events and assign scores as per their performance that helps in issuing certificates to the participants.

3.2.2 Sequence diagram

A sequence diagram is a Unified Modeling Language (UML) diagram that illustrates how a group of objects interact and operate with each other sequentially. Here all the sequence diagrams show how each actor interacts with the interface and explains the outcome of each action.

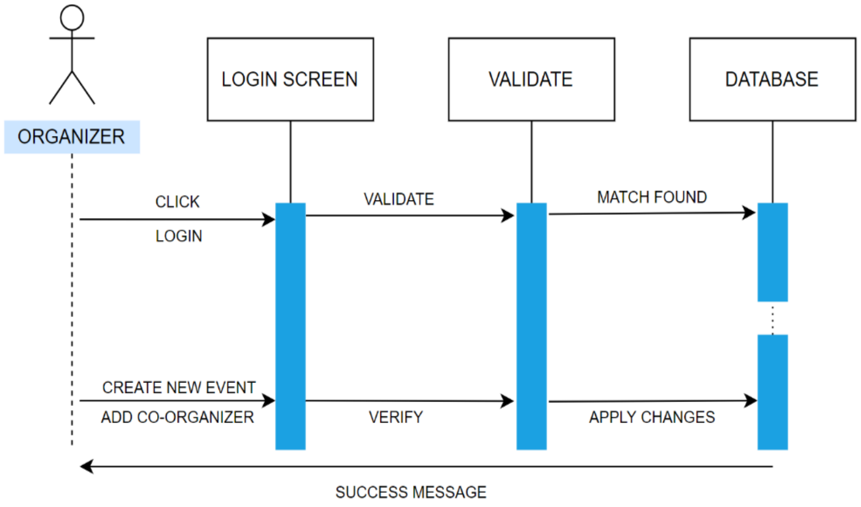


Fig. 3.2.2.1 Organizer Sequence Diagram

The organizer first clicks the login button that redirects him to the login page where the organizer enters his credentials. Once the credentials are validated, he gets access to the dashboard where he has the option to create and view events and add co-organizers to manage the events. Once all the details are verified and submitted a success message will be issued.

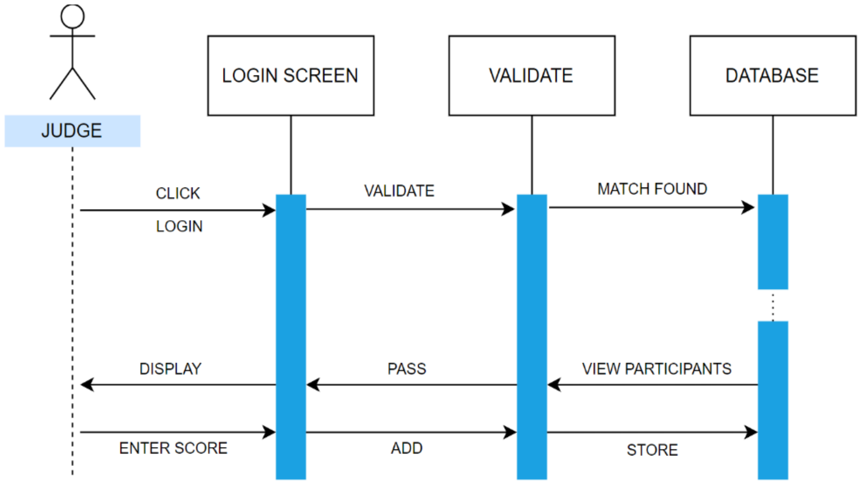


Fig. 3.2.2.2 Judge Sequence Diagram

The judge first clicks the login button that redirects him to the login page where he enters his credentials. Once the credentials are validated, he gets access to the dashboard where he has the option to view all the details of the event and once the event begins, he has the option to enter the scores of the participants on basis of their performance that are stored in the database and used to issue certificates to the participants accordingly.

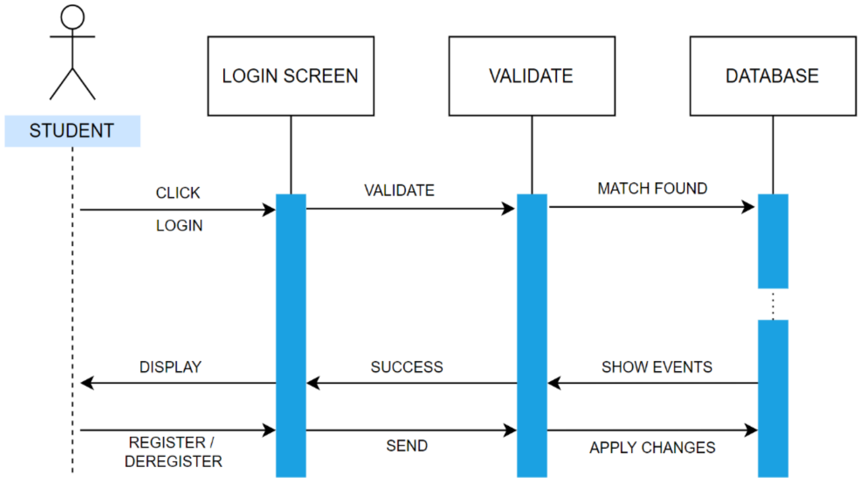


Fig. 3.2.2.3 Student Sequence Diagram

The student first clicks the login button that redirects him to the login page where he/she enters his/her credentials. Once the credentials are validated, he/she gets access to the dashboard where they have the option to view all the events and register for the events of their interest. Once it’s done the registration of each participant is stored in the database.

3.3 Functional Design

3.3.1 Complete system flow diagram

The system flow diagram shows the activities carried out by each actor using the interface. Organizer adds all the events and assigns one or more Co-Organizers to each event to handle them. Co-Organizer will be managing a particular event and will have access to only that event. He will be responsible for the event he is conducting and will be able to publish results of the same. Students will basically be able to register for an event and view the details of that event. He/she will also be even given an option to cancel the event if necessary. Judge will judge the particular event or game and allot the scores that will be used to declare the winner.

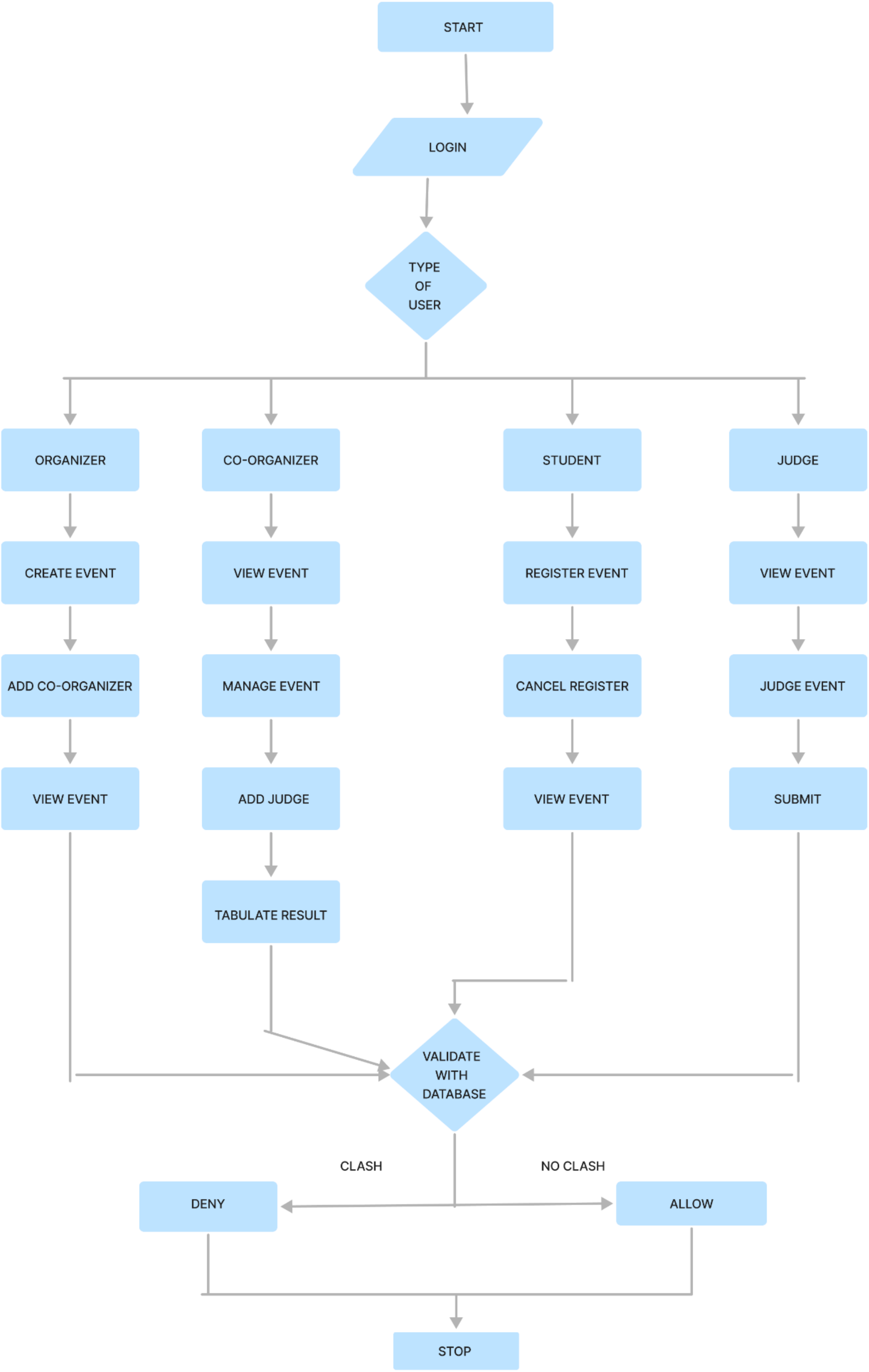


Fig. 3.3.1 System Flow Diagram

3.4 Access layer Design

3.4.1 Database Schema

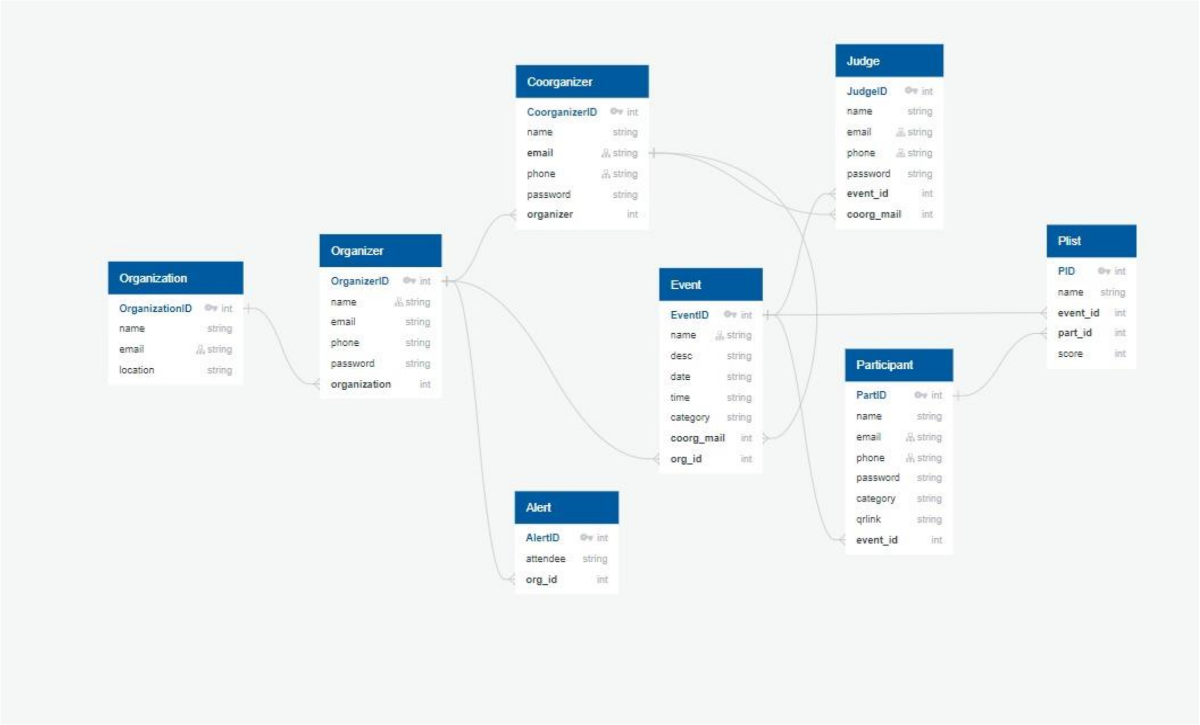


Fig. 3.4.1 Database Schema

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. The diagram shows how data is stored in a relational database structured with seven different tables namely Organizer, Co-Organizer, Judges, Events, Participants, Scores and Certificates and all the fields associated with each table. It also depicts the relation between tables and constraints imposed on each field.

CHAPTER 4

IMPLEMENTATION

Chapter 4

IMPLEMENTATION

4.1 Software Used with justification

Frontend Technology

HTML: It helps in structuring web pages, navigating images and videos, interacting with API's that are displayed on Internet.

CSS: It is responsible for the aesthetics and the visual appeal of a page. And also helps in Controlling Effects and Flash Animation, managing the Web Templates etc.

JavaScript: It helps in creating dynamic and interactive web content like applications and browsers. Where HTML and CSS are languages that give structure and style to web pages, JavaScript gives web pages interactive elements that engage a user.

Bootstrap: It is an open source, sleek, intuitive, and powerful, mobile first front-end framework for faster and easier web development.

Sweetalert: It is a method to customize alerts in your applications, websites and games. It allows the user to change it with a standard JavaScript button.

Backend Technology

Flask: It is a small and lightweight Python web framework that provides useful tools and features that make creating web applications in Python easier.

SQLite: It lets you store data in structured manner. SQLite has higher performance. SQLite databases can also be queried and the data retrieval is much more robust

CHAPTER 5

TESTING

CHAPTER 5

TESTING

5.1 Test Objectives

Software testing is the process of investigating, verifying and validating software or applications to ensure that they are bug free and providing the stakeholders with information related to the quality of the software or service being tested. It gives the business a wider perspective to appreciate and understand the risks involved in software implementation.

Two steps are involved in software testing, primarily the verification and validation of properties. These characteristics typically reveal the extent to which the system being tested satisfies the criteria that guided its design and development, whether the system responds accurately to vivid inputs, completes its tasks within a reasonable amount of time, is simple to use, can be installed and run in intended environments, and overall, whether it produces the desired outcome for its stakeholders.

In order to detect software faults and enhance accuracy and efficiency, software testing often involves running a programme or application. Testing is a procedure that is carried out through iterations since when one bug is fixed, it could reveal another.

5.2 Types Testing Conducted

5.2.1 Unit testing

In Unit testing, individual units of software are tested. Unit testing is carried out to validate each unit of the software and its performance. This type of testing is usually done by developers on the go, to ensure that each unit is working and functioning as anticipated.

In procedural programming, a unit could be an entire module, but it is more commonly known as an individual function or procedure. Unit testing is performed by White Box testing, where the testing is based on an analysis of the internal structure of the component or system. Unit tests are short code fragments created by programmers or occasionally by white box testers during the development process. It forms the basis for components testing. Ideally, each test harness can be used to assist testing a module in isolation. Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended.

Table 1: Test cases for Unit Testing

No.	Test case	Expected Outcome	Observed Outcome
1	Participant register button is clicked	The participant must be redirected to participant dashboard	The participant is redirected to his dashboard
2	Organizer creates a new event	New event must be added to list of all events	A new event is created
3	Co-organizer edits an event description	Description must be changed	Co-organizer can change the description

5.2.2 System testing

System testing is the process of testing the entire integrated system as a whole. This test is carried out to evaluate the system’s end-to end compliance with the stated requirements. System testing includes black-box testing, where an individual need not have knowledge of the inner design of the code or logic. Black box testing is well suited for large code segments. System testing is performed after integration testing and unit testing. The purpose of integration testing is to detect any inconsistencies between the application units that are integrated together (called assemblages) or between any of the defects both within the “inter-assemblages” and also within the system as a whole.

Table 1: Test cases for System testing

No.	Test Case Description	Expected Outcome	Observed Outcome
1	Organizer can create event and assign co-organizer	New event should be created and co-organizer will should be notified	Co-organizer was notified through mail
2	Participant register for event and it will be displayed to co-organizer	Co-organizer should have the list of all participants	Co-organizer was able to view the list of participants
3	Email notification for guests	Guests can be notified through email with single click	Guests receives notification through mail
4	Email notification for judge	When a judge is assigned to event, we should be notified through mail	Judge receives a notification when he is assigned any event
5	Providing participation certificate in bulk	When co-organizer uploads csv, certificated should be distributed to all	Certificated was distributed to everyone with the upload of csv provided in app

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

The proposed Event Management System is to automate the existing manual system by the help of computerized equipment and full-fledged computer application fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. The required software and hardware are easily available and easy to work with.

The purpose of the project is to build an application program to reduce the manual work for managing the Event, Activity, Payment, Organizers. It tracks all the details about the Organizers, Attendees, Conductors. This also provides an automatic certificate of participation and winner certificates in one single app. The interface in turn helps in generating and mailing of certificates too.

Our application makes it simple to organize an event and manage it, as well as track progress on the go, a completely seamless event on the go. Data from participants and events are better visualized on the dashboard. All Integrated judgement within the web app has an automated certificate. Overall, this application minimizes the amount of extra work that event organizers have to do when administering a large-scale event.

Few events which can be conducted through mobile can be integrated into this application. Adding map location of each event can make it easy for participants to reach the event spot.

REFERENCE

REFERENCES

[1] https://docs.python.org/3/

[2] https://flask.palletsprojects.com/en/2.1.x/

[3] https://getbootstrap.com/docs/4.1/getting-started/introduction/

[4] https://sweetalert.js.org/docs/

[5] https://docs.sqlalchemy.org/

[6] https://developer.mozilla.org/en-US/docs/Web/JavaScript

[7] https://github.com/Asabeneh/30-Days-Of-JavaScript

[8] https://docs.github.com/en

[9] https://www.sqlite.org/docs.html

[10] https://devdocs.io/css/

[11] https://www.dpc.nsw.gov.au/tools-and-resources/event-starter-guide/documentation/

[12] Glenn Bowdin, Johnny Allen, William ÓToole, Rob Harris, Ian McDonnell, Events Management, Routledge London and New York 2011.

[13] Silvers, Julia Rutherford, Risk Management for Meetings and Events, Elsevier Amsterdam 2011.

[14] R.A. Khan, Extending and Harmonizing Knowledge Areas and Process Groups in PMBOK, Dortmund 2009, EuroMPM Master Thesis.

[15] Oliver Thomas, Bettina Hermes, Peter Loos, "Reference modelbased event management," International Journal of Event Management Research, Volume 4, Number 1, 2008.

[16] Peter J. A. Reusch, "Project management supported by object role modeling, international conference on new challenges of economic and business development," University of Latvia, Riga 2012, http://www.evf.lu.lv/ conf2012/

APPENDIX

APPENDIX

Appendix 1: Screenshots of the Application

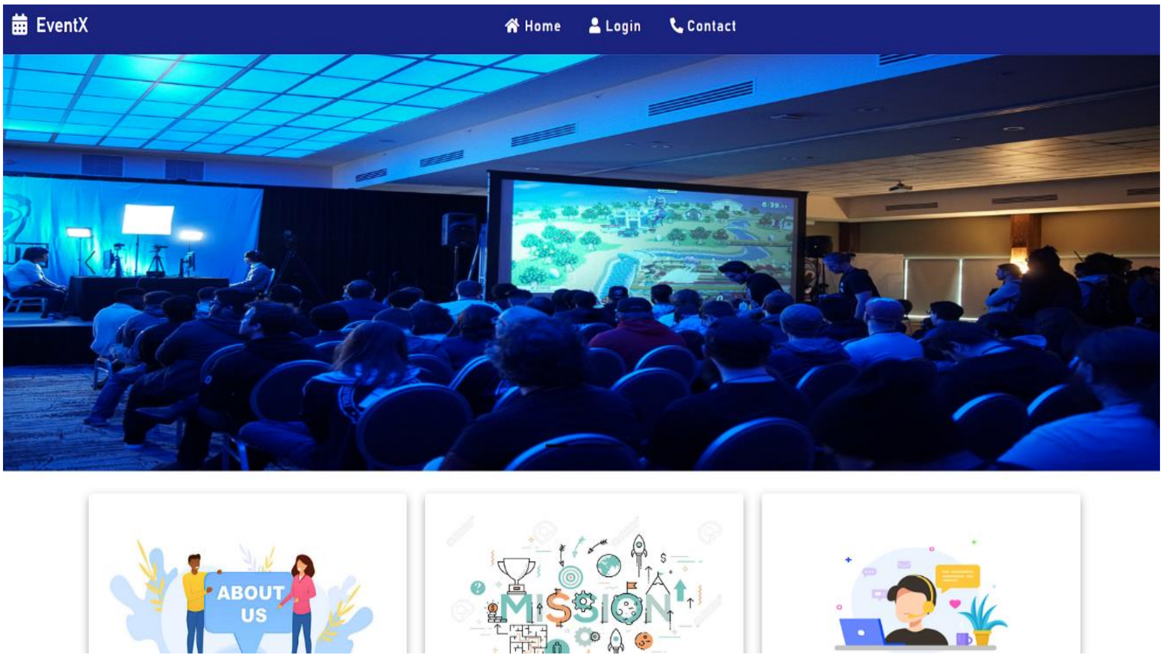


Fig. 1 Main dashboard

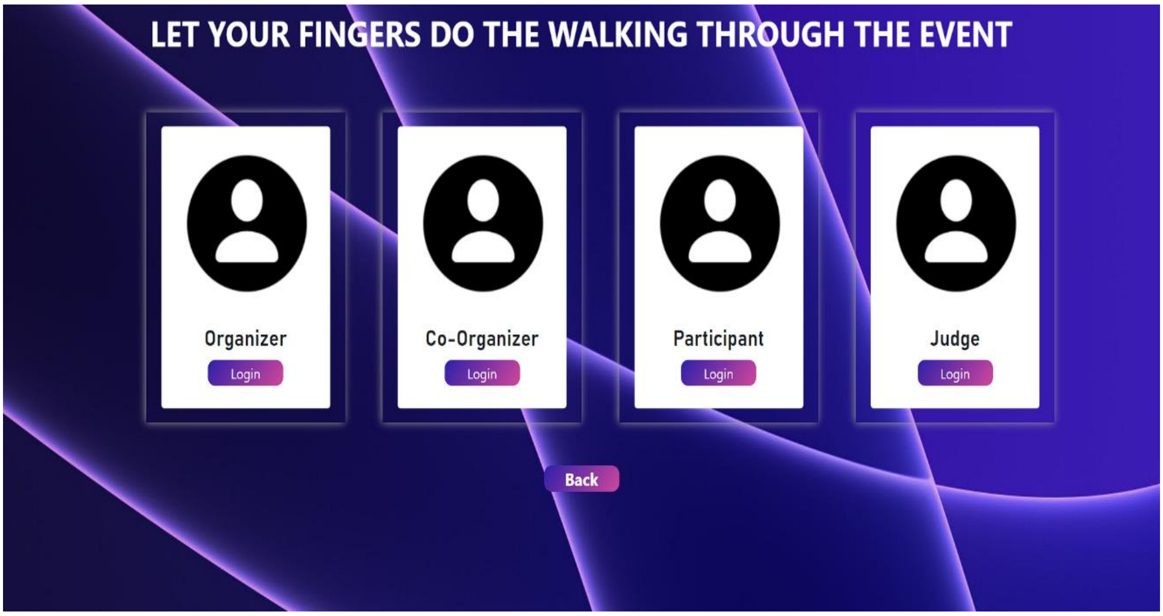


Fig. 2 Role selector

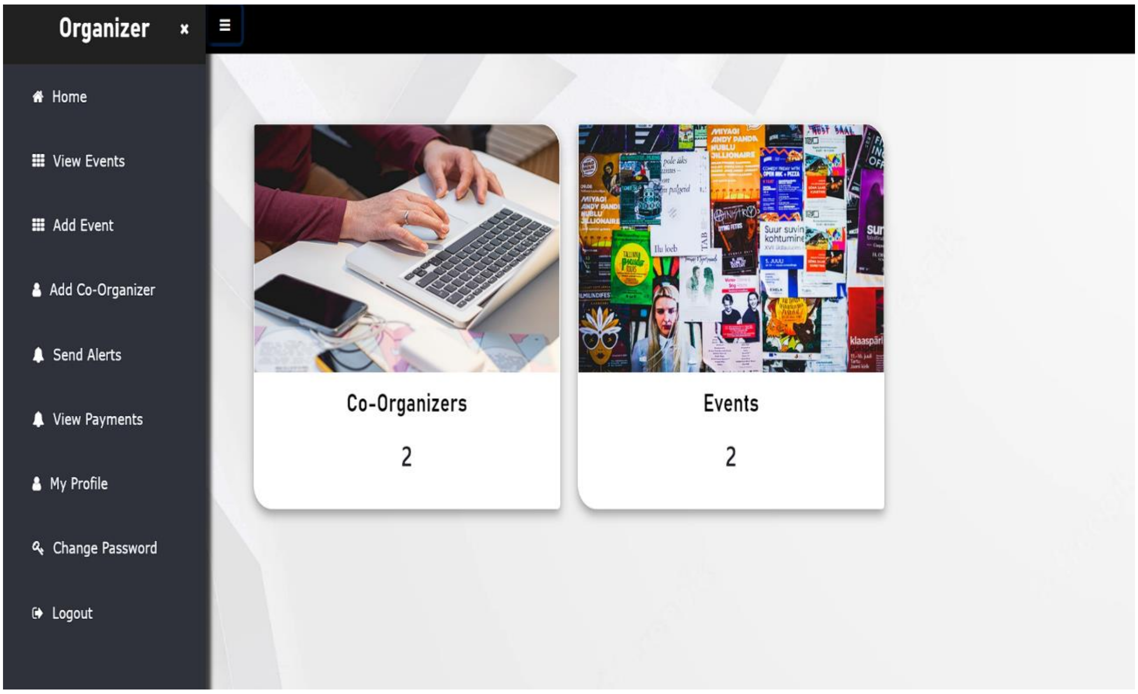
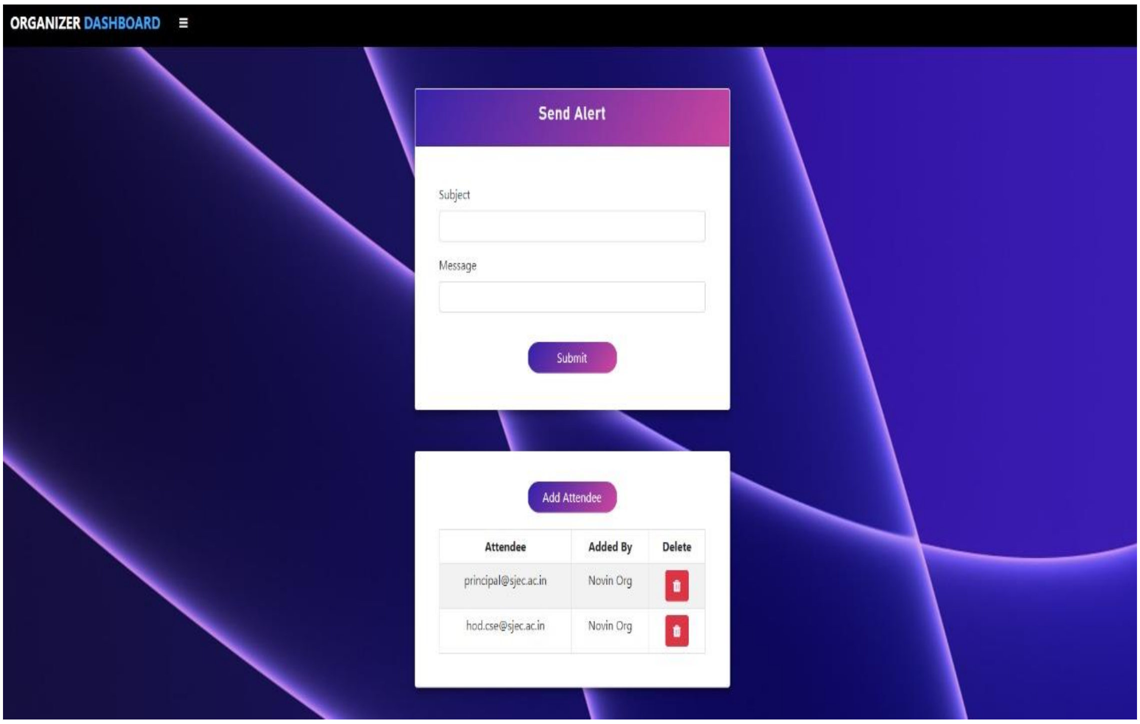
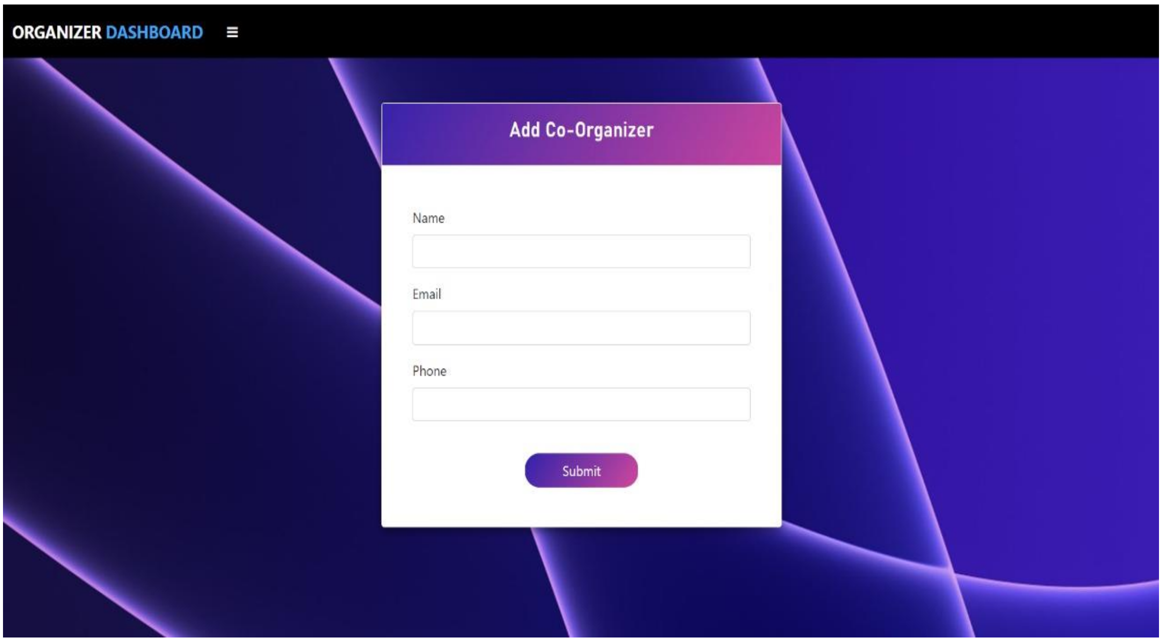


Fig.3 Organizer Dashboard



Attendee	Added By	Delete
principal@sjec.ac.in	Novin Org	🗑
hod.cse@sjec.ac.in	Novin Org	🗑

Fig.4 Send alert to attendee



ORGANIZER DASHBOARD ≡

Add Co-Organizer

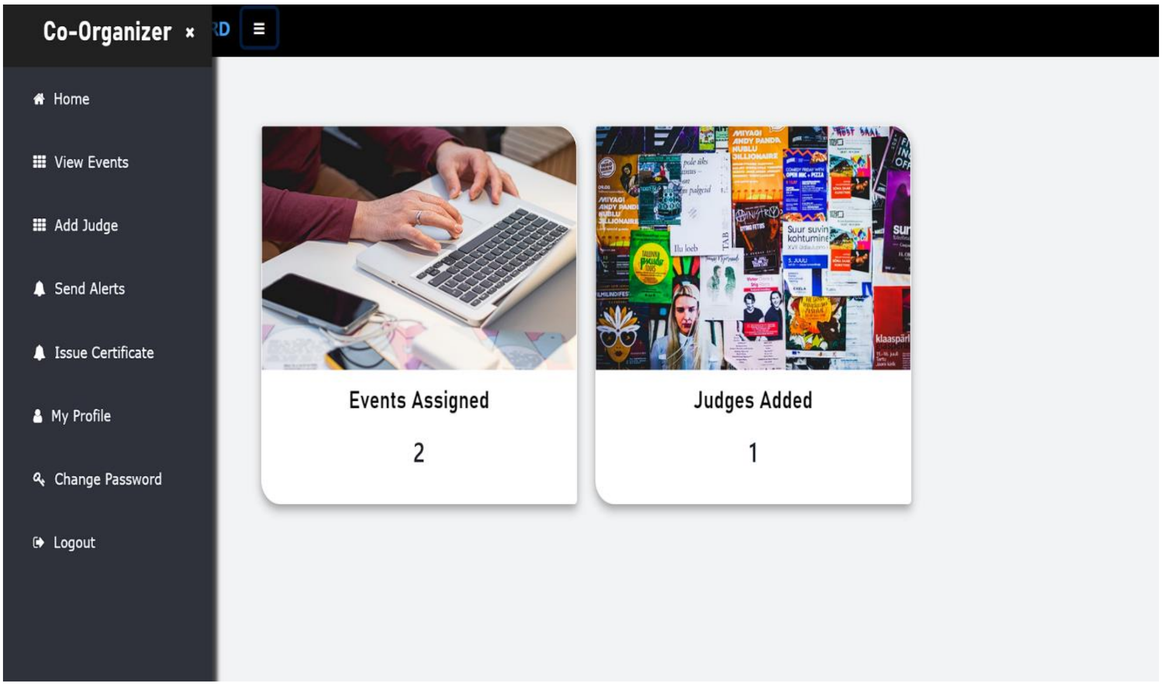
Name

Email

Phone

Submit

Fig. 5 Organizer assign Co-organizer



Events Assigned

2

Judges Added

1

Fig.6 Co-organizer assign judge

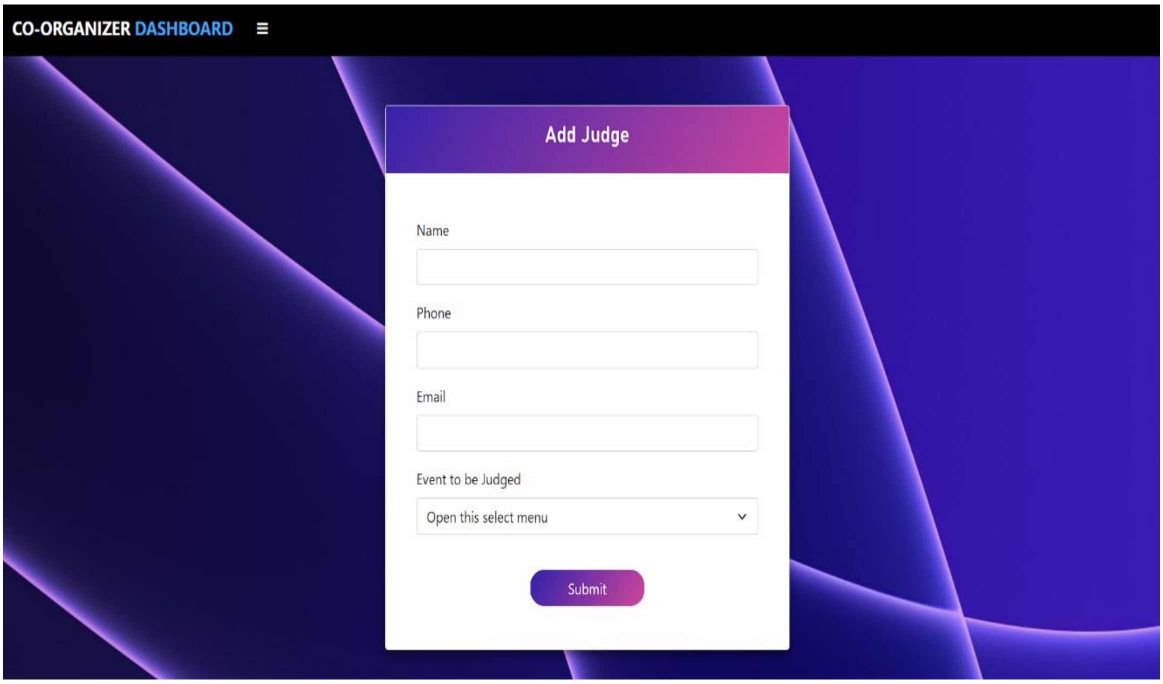
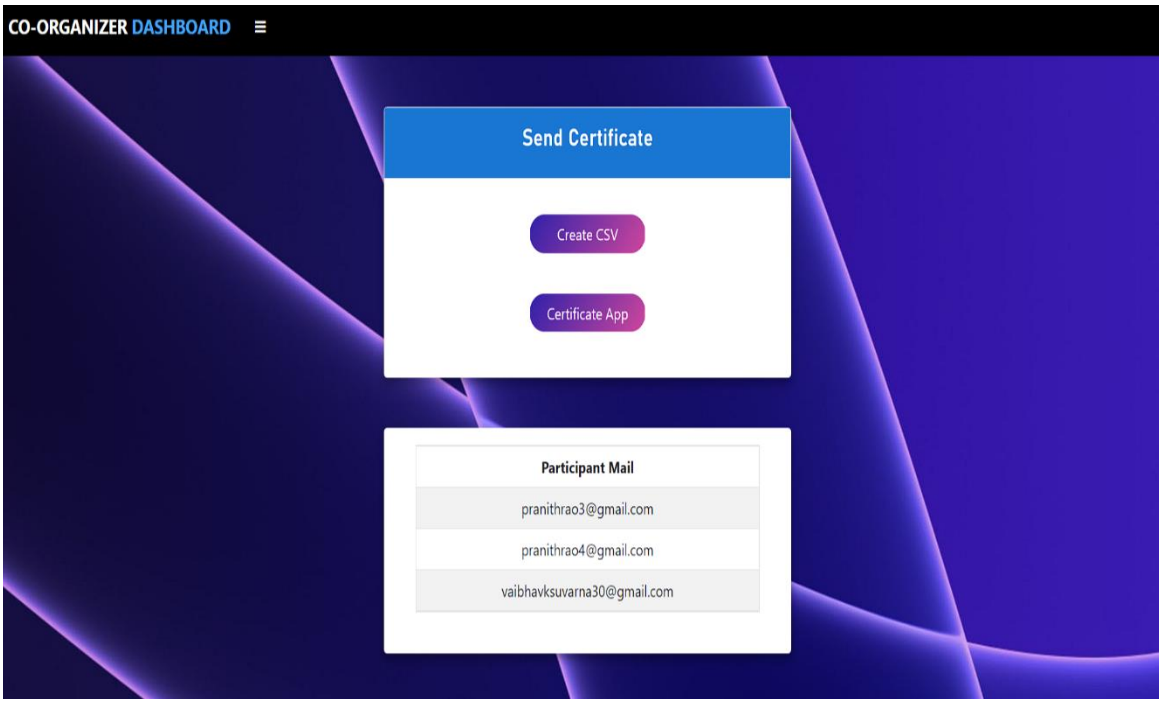
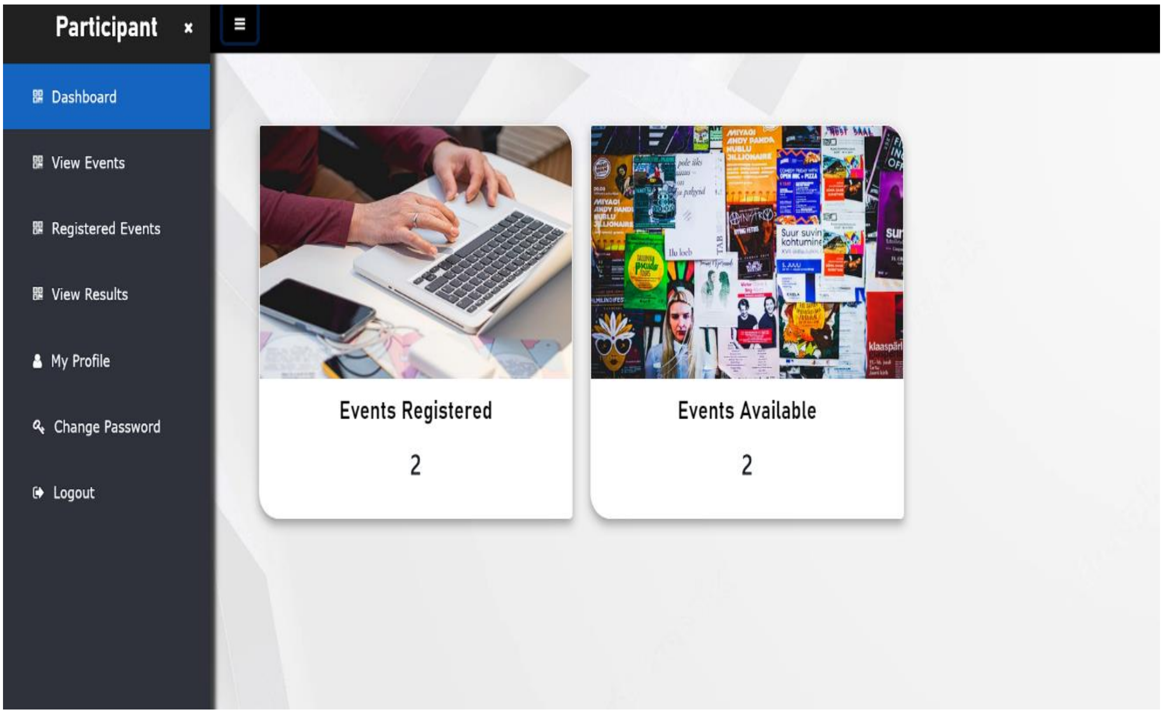


Fig. 7 Co-organizer assign judge



Participant Mail
pranithrao3@gmail.com
pranithrao4@gmail.com
vaibhavksuvarna30@gmail.com

Fig.8 Send Certificate to participant



Events Registered

2

Events Available

2

Fig. 9 Participant dashboard

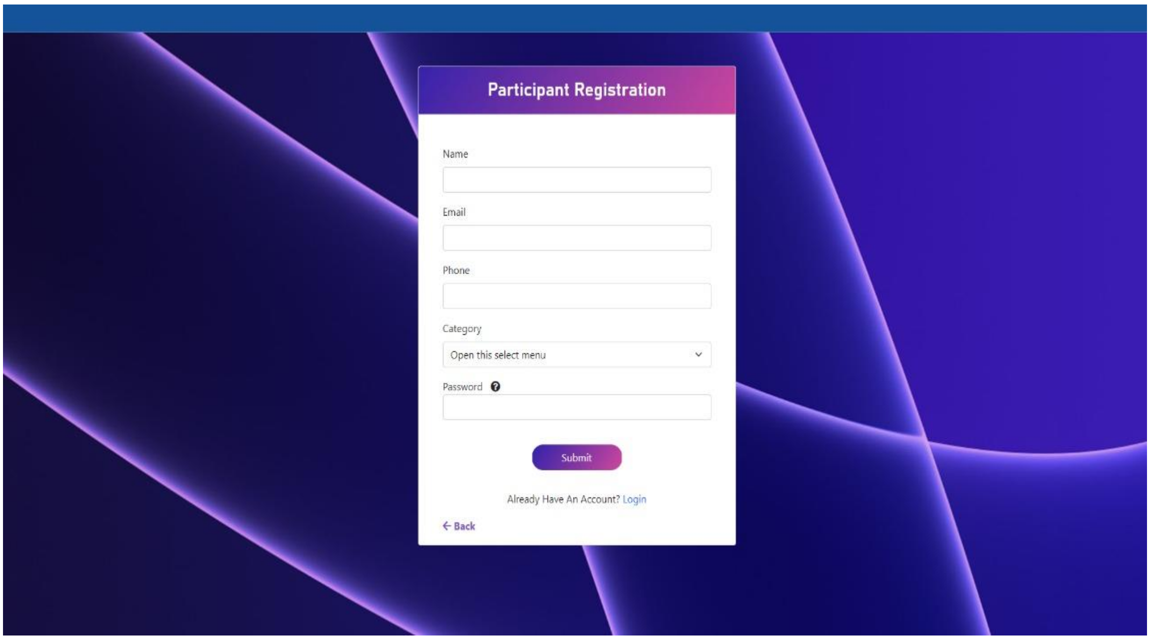
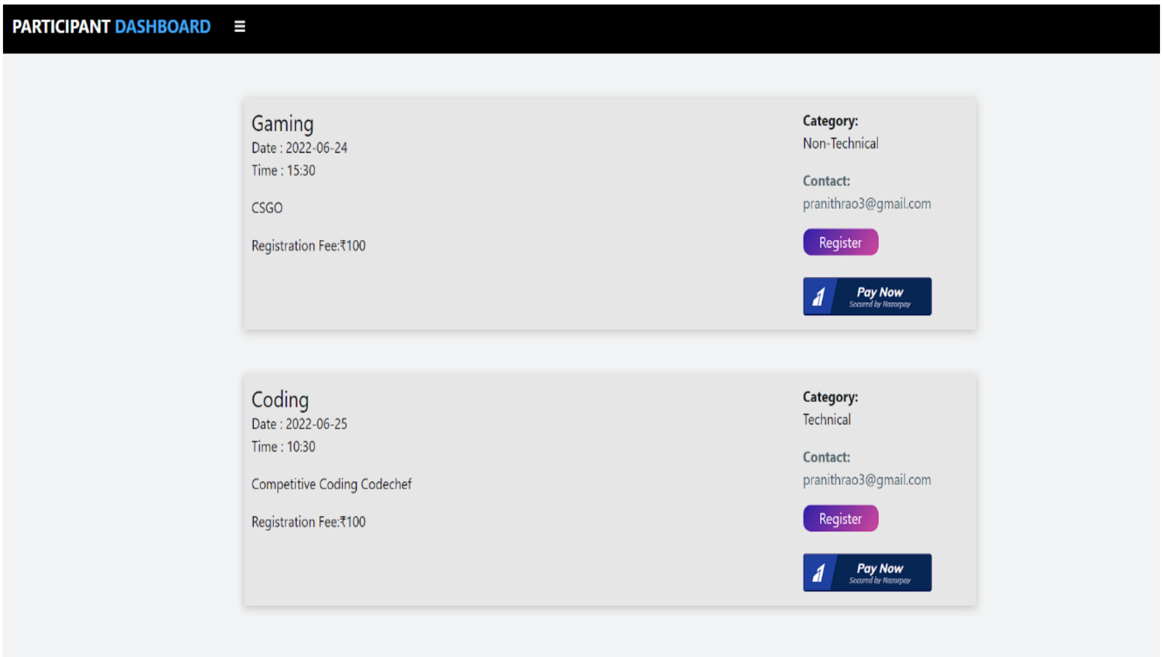


Fig.10 Participant registration



Gaming
Date : 2022-06-24
Time : 15:30

CSGO

Registration Fee:₹100

Category:
Non-Technical

Contact:
pranithrao3@gmail.com

Coding
Date : 2022-06-25
Time : 10:30

Competitive Coding Codechef

Registration Fee:₹100

Category:
Technical

Contact:
pranithrao3@gmail.com

Fig. 11 Participant view events

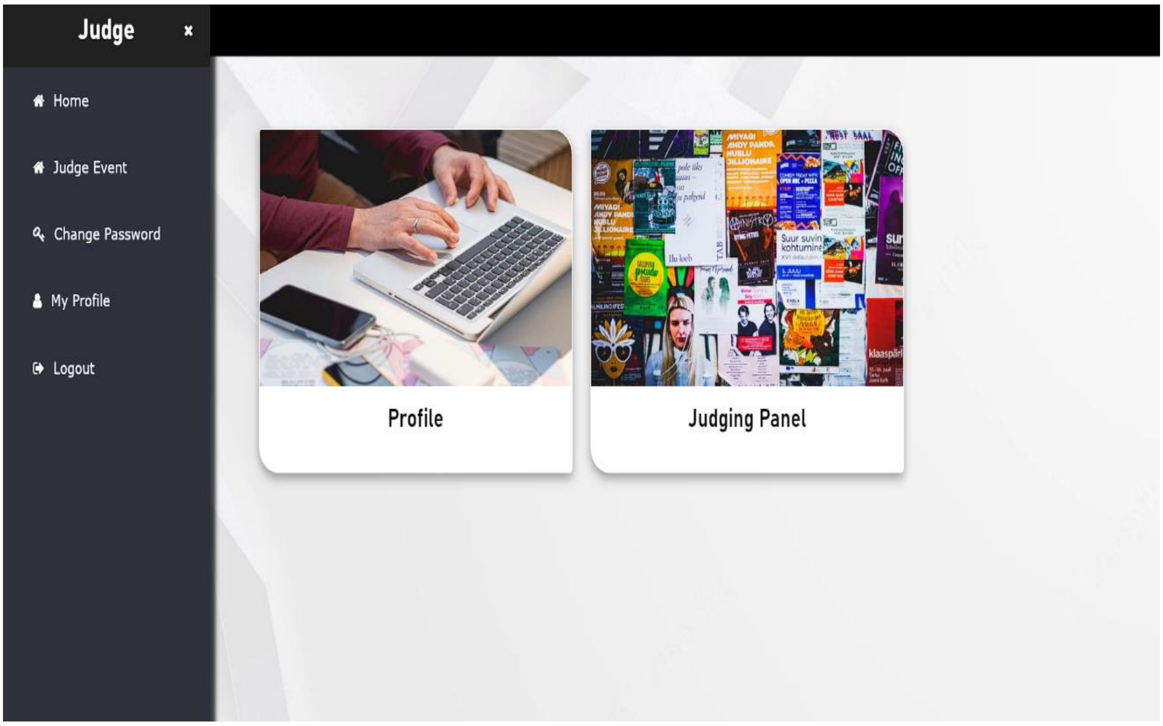
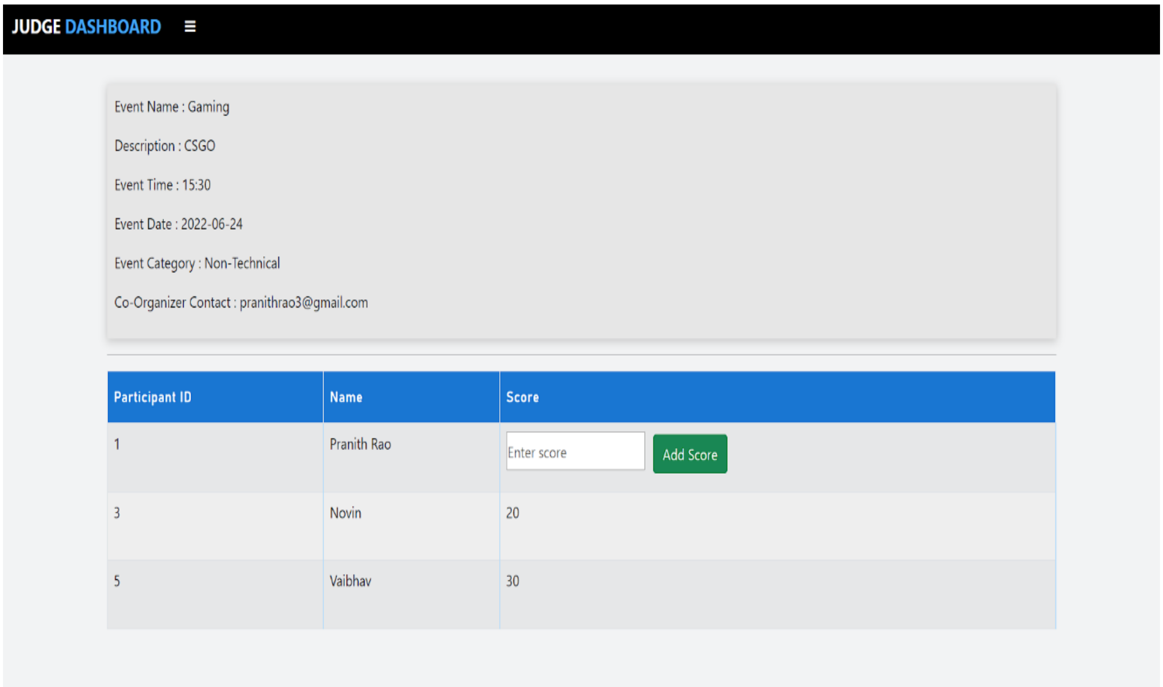


Fig. 12 Judge dashboard



Event Name : Gaming

Description : CSGO

Event Time : 15:30

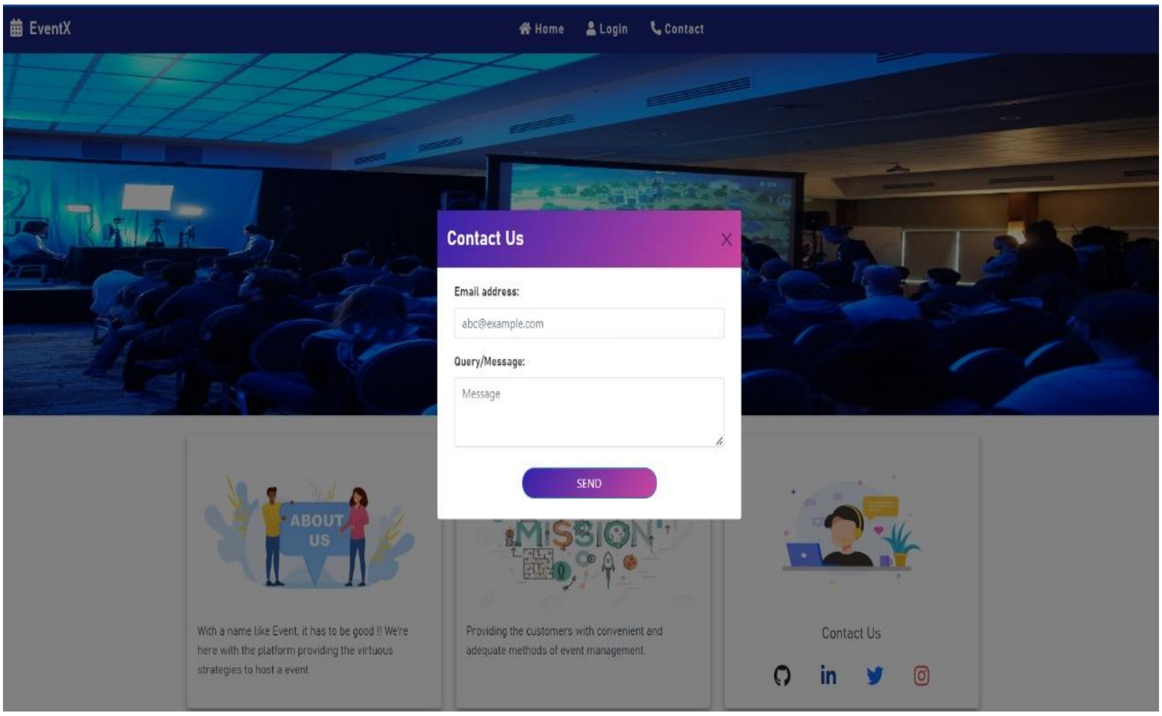
Event Date : 2022-06-24

Event Category : Non-Technical

Co-Organizer Contact : pranithrao3@gmail.com

Participant ID	Name	Score
1	Pranith Rao	Enter score Add Score
3	Novin	20
5	Vaibhav	30

Fig. 13 Judge enter marks



Contact Us

Email address: abc@example.com

Query/Message: Message

With a name like Event, it has to be good !! We're here with the platform providing the virtuous strategies to host a event.

Providing the customers with convenient and adequate methods of event management.

Contact Us

Fig.14 Contact Form