VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANASANGAMA, BELGAVI-590018

DATABASE MANAGEMENT SYSTEM MINI PROJECT REPORT (17CSL58)
ON

“EVENT REGISTRATION AND SURVEY”

Submitted in partial fulfillment of the requirements for the 5th Semester
Information Science and Engineering

Submitted by

RAJAT JAIN
1BI17IS034

Under the guidance of

Mr. PADMANABHA J
Assistant Professor
Dept. of ISE, BIT
Bangalore-04

2019-2020

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING
BANGALORE INSTITUTE OF TECHNOLOGY
K. R. Road, V. V. Puram, Bengaluru-560004

BANGALORE INSTITUTE OF TECHNOLOGY

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

K. R. Road, V. V. Puram, Bengaluru-560004

CERTIFICATE

This is to certify that the DBMS mini-project entitled **“EVENT REGISTRATION AND SURVEY”** is a bonafide work carried out by **Mr. RAJAT JAIN (1BI17IS034)** in a partial fulfilment for the award of Degree of Bachelor of Engineering in **Information Science and Engineering** of the university **Visvesvaraya Technological University, Belgaum** during the academic year 2019-20

Lab In charge 1:

Mrs. Anupama K C
Asst Professor
Dept of ISE
BIT, Bangalore

Lab In charge 2:

Mr. Padmanabha J
Asst Professor
Dept of ISE
BIT, Bangalore

Signature of the HOD
Dr. J. Prakash
HOD
Dept. of ISE
BIT, Bangalore

Name of the Examiners **Signature with date**

1.

2.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible. So, with gratitude, I acknowledge all those whose guidance and encouragement crowned my effort with success.

It's my pleasure in thanking my principal Dr. Aswath M.U, BIT, Bangalore, for providing an opportunity to present this project as a part of my curriculum in the partial fulfillment of the degree.

I express my sincere gratitude for Dr. J. Prakash, Head of the Department, Information Science and Engineering for his co-operation and encouragement at all moments of my approach.

It is my pleasant duty to place on record my deepest sense of gratitude to my guide Mrs. Anupama K.C, Asst. Professor, for the constant encouragement, and Mr. Padmanabha J, Asst. professor for valuable help and assistance in every possible way.

I would like to thank all my teachers and non-teaching staff of ISE Department for providing me their valuable guidance and for being there with me at all the stages of my word.

RAJAT JAIN
1BI17IS034

ABSTRACT

Event registration and survey is an effort to allow the users to register for a given event. With event registration and survey students can register for an event online or give a survey regarding a preattended event in their own time regardless where they are. You need a browser and an working internet connection.

The purpose of the system is to develop an online event registration and survey system used to simplify the procedure for registering for an event manually via pen and paper method. The manual procedure used for conducting events registration is time consuming and error prone.

This is a web based application that allows the students to allow the users to register for an event online. In this paper, the proposed system was developed using Hyper Text Markup Language (HTML), Python, Structured Query Language (SQL), Cascading Style Sheet (CSS) and JavaScript.

Table of Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

1.1 General Overview

A Database Management System (DBMS) is system software for creating and managing databases. A database is a collection of information that is organized so that it can be easily be accessed, managed and updated. The DBMS provides users and programmers with a systematic way to create, retrieve, update and manage data. Most of giant IT farms are investing billions of business information of raw data manage them in files and documents which may sometimes even suffer loss hence no security or backup. This project makes a decent effort to overcome this problem.

1.2 Problem Statement

Event Registration is very helpful to Students. The aim of this project is to provide quick, immediate and easy way for the students to register for an event. It can provide special advantages to students. The Event registration system can help the students to register for any event they are interested in and later provide their views on the event through a survey form. The login module helps the admin or student to login to the site's respective section. For that, he/she must type the username and password correctly. The login provision helps the already registered user to directly access the site. Student section is mainly for students. This helps the students to register for the event or give a survey for a previously attended event. The system keeps the responses of the students in the database. Administrator module is mainly for the Staff. This will contain the creation and deletion of Events, create or delete a student,view the students and add a survey to a particular subject. The admin can view all his/her pervious results for the exams already taken in the results section.

1.3 Objectives

- To make the registration System simpler, efficient and errorless.
- To create strong and secrete database that allows for any connection in a discrete way, to prevent any outside or inside attack.

- To save time consumption by manual process of event registration and survey.
- To register for an event from anywhere at any time with proper internet access.
- The registration is done instantly.

1.4 Purpose

Event Registration and Survey System is a web application which aims at computerization of current procedure of Event Registration and Survey. Currently, the process involves printing the hardcopy of Registration and Survey forms and providing them to the students and receiving their responses manually. This takes more time and is less efficient. The main aim of the project circles around the fact to make registration and surveying process by the students easier and more reliable with lesser effect of errors caused mainly by humans.

1.5 Scope

- Event Registration and Survey is designed for schools and colleges.
- The student can see the list of events available to be registered.
- The student can choose one of them and register for an Event online that will be conducted by the concerned department.
- The student can provide survey for the event that he/she has attended.
- The staff can check the result of the survey.

The web application has two parts:

- User End (Students)
- Admin End (School/College Staff)

The students can register the event that they are interested in and provide survey for the event that they would have attended.

The admin keeps the track of all the events that are currently being held by the concerned department and also view the response provided by the users to the survey form created by the admin.

CHAPTER 2

SOFTWARE AND HARDWARE REQUIREMENTS

2.1 Software Requirements

NAME OF THE COMPONENT	SPECIFICATION
Operating System	Windows 10
Language	HTML, CSS, Python, JavaScript
Database	PostgreSQL
Browser	Any of Mozilla, Google Chrome, Opera etc.
Web Server	Python
Software Development Kit	Django
Scripting Language enable	HTML, Python, CSS
Database Connection	Django

Table 2.1 Software Specification

2.2 Hardware Specification

NAME OF THE COMPONENT	SPECIFICATION
Processor	Core i5 processor
RAM	8GB
Hard Disk	1TB
Monitor	14’color monitor
Keyboard	122 Keys

Table 2.2 Hardware Specification

2.3 Front End Tool

Hypertext Mark-up Language (HTML) is the standard mark-up language for creating web pages and web applications.

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a mark-up language like HTML

HTML is used to create the actual content of the page, such as written text, and CSS is responsible for the design or style of the website, including the layout, visual effects and background color.

The Hypertext Mark-up Language (HTML) is used in almost all the files we’ve used. This is used to create the interface on which the whole system works.

The Cascading Style Sheets (CSS) is used in all the files where there is a need to display the interface making it more presentable and creative

2.4 Back End Tools

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Django is a framework of python which is used for web development of applications which can be further deployed into the vast internet world to be accesible by the users all across the globe and providng them a better web services. Django is a easy framework which can be learnt by any regular person.

- Establishing the connection to the Database
- Requesting data from the Database through SQL queries
- Retrieving the requested data from the Database
- Displaying the retrieved data

CHAPTER 3

DESIGN

3.1 Entity Relationship Diagram (ER Diagram)

An entity-relationship diagram (ERD) is a data modelling technique that graphically illustrates an information system's entities and the relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure.

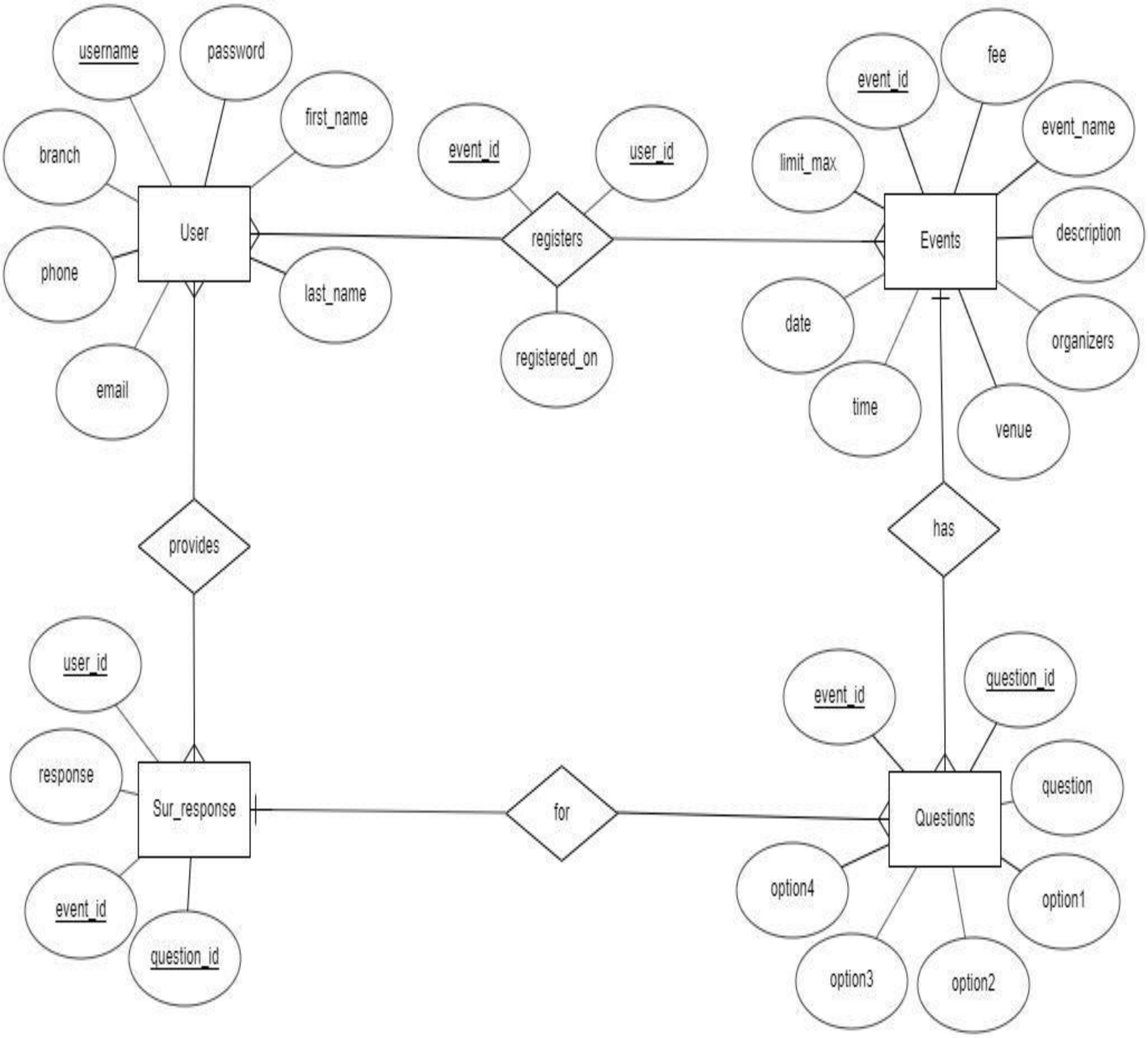


Figure 3.1 E. R representation of the Project

3.2 Schema Diagram

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. It formulates all the constraints that to be applied on the data.

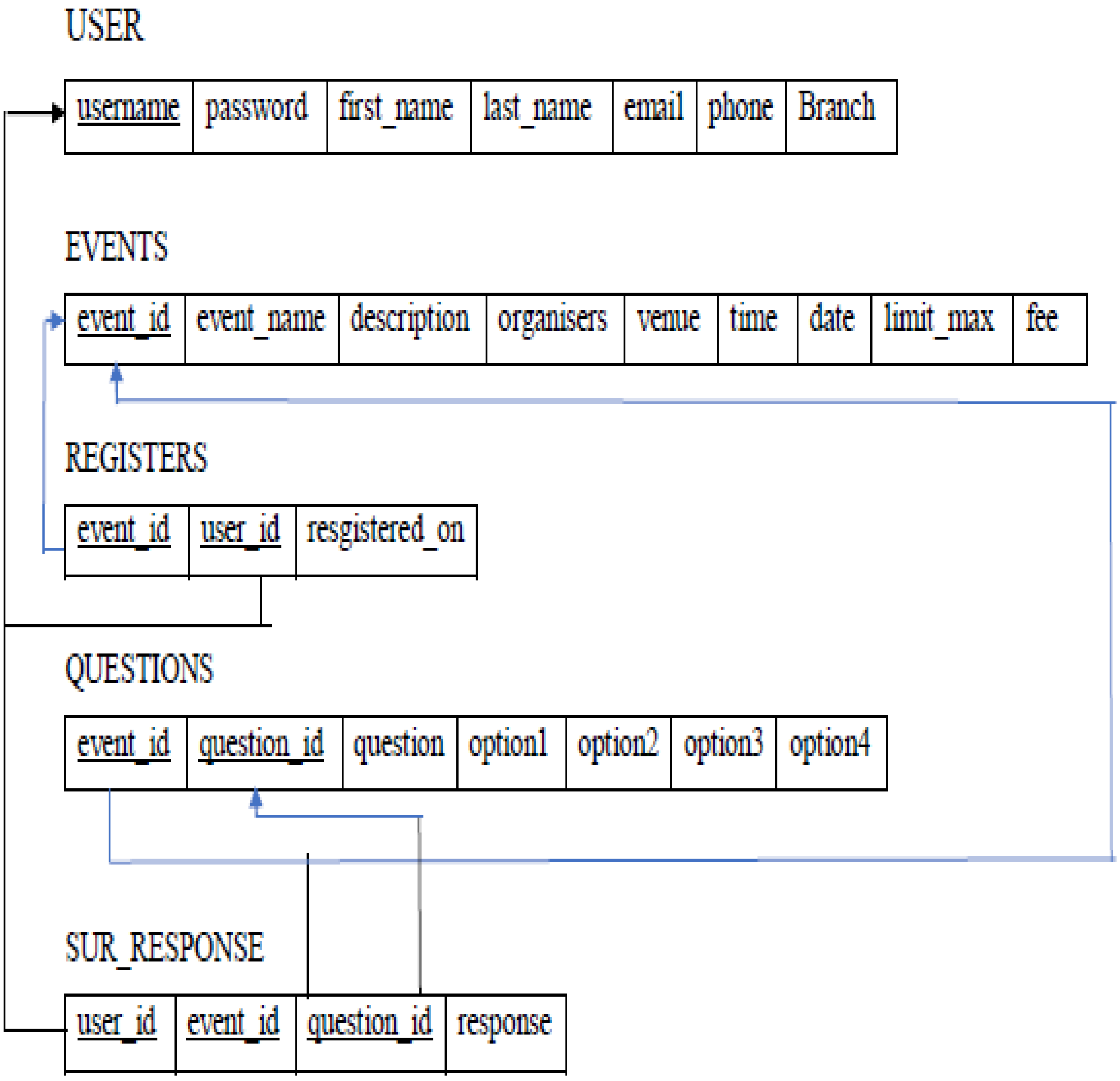


Figure 3.2 Schema representation of the Project

3.3 Entities Attributes and Relationships

An entity may be an object with a physical existence (for e.g. A particular person, car or employee) or it may be an object with a conceptual existence (for e.g. company, a job or a university course)

Each entity has attributes i.e. the particular properties that describe it. The attribute values that describe each entity become q major part of the data in the database. When an attribute of one entity type refers to another entity type, a relationship exists.

This section of the document explains the entities used in the project, heir attributes and how they will work together. Basically, this is intended to make the design easier and more understandable for everyone.

Entities and Relationships

1. **Users**
   This table has students information. User can fill out a survey form for the events they have attended. Here a unique username is given to each student.

2. **Events**
   This table has all the information about the events going on in the College. Here, all the events are stored which are ought to be displayed as per the choice of event made by user.

3. **Question**
   This has questions for the event survey attended by students. These are the available set of questions pre created by the admin for which users can fill out a survey form.

4. **Sur_response**
   This table has the response given by the students for an event. This is used to keep track of the responses provide by the users

5. **Registered**
   This table has the students who have registered for a particular event.

3.3.1 Table: Users

This table has students information.

They have the following relationship.

1. Users sign up for the events.
2. From the available events students register for the event they are interested.

User can fill out a survey form for the events they have attended. Here a unique username is given to each student. Username is the primary key in this table where all the details of the students is stored i.e. his/her full name email id and phone number. Username is unique to each student hence cannot be repeated.

Attributes

Name	**Data Type**	**Type**
username	varchar	Primary Key Attribute
password	varchar	Non-Key Attribute
first_name	varchar	Non-Key Attribute
last_name	varchar	Non-Key Attribute
phone	varchar	Non-Key Attribute
branch	varchar	Non-Key Attribute
email	email	Non-Key Attribute

Table 3.1 Users attributes

3.3.2 Table: Events

These are list of events and their corresponding details.

They have the following relationship

1. A user can register for a particular event.
2. When the user selects an event, all the details of that event are displayed.

Here, all the events are stored which are ought to be displayed as per the choice of event made by user.

Attributes

Name	**Data Type**	**Type**
event_id	Varchar	Primary Key Attribute
event_name	Varchar	Non-Key Attribute
description	varchar	Non-Key Attribute
organizers	varchar	Non-Key Attribute
venue	varchar	Non-Key Attribute
time	Time	Non-Key Attribute
date	Date	Non-Key Attribute
limit_max	Varchar	Non-Key Attribute
Fee	Varchar	Non-Key Attribute

Table 3.2 Events attributes

3.3.3 Table: Questions

These are the available set of questions pre created by the admin for which users can fill out a survey form. Here, event_id and question_id constitute a composite primary key.

The relationships they have are:

1. Student selects a particular event whose survey he/she wants to fill out.
2. The questions with respect to that event are displayed.

Attributes:

Name	Data Type	Type
question_id	Int	Primary Key Attribute
event_id	Varchar	Foreign-Key Attribute
question	Varchar	Non-Key Attribute
option1	Varchar	Non-Key Attribute
option2	Varchar	Non-Key Attribute
option3	Varchar	Non-Key Attribute
option4	Varchar	Non-Key Attribute

Table 3.3 Questions attributes

3.3.4 Table: Sur_response

To keep track of the responses provide by the users. Here, event_id, user_id and question_id constitute a composite primary key.

The relationships it has are:

1. When the user attends a survey, the corresponding responses are stored in this table.
2. A user provides a single response for every question.

Attributes:

Name	Data Type	Type
event_id	Varchar	Foreign-Key Attribute
user_id	Varchar	Foreign-Key Attribute
question_id	Varchar	Foreign-Key Attribute
response	Int	Non-Key Attribute

Table 3.4 Sur_response attributes

3.3.5 Table: Registered

These are the entries to keep track of the students that have registered for any of the available events. Here, event_id and user_id constitute a composite primary key.

The relationships it has are:

1. User logs in the system and registers for an event.
2. The respective event_id and user_id are stored in this table.

Attributes

Name	Data Type	Type
event_id	Varchar	Foreign-Key Attribute
user_id	Varchar	Foreign-Key Attribute
registered_on	Date	Non-Key Attribute

Table 3.5 Registered attributes

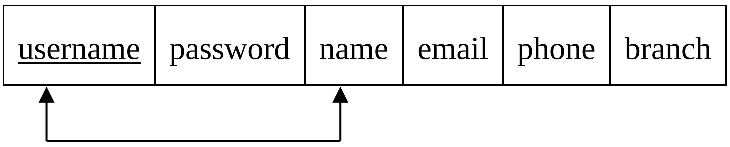
3.4 Normalization

3.4.1 FIRST NORAMAL FORM (1NF)

First Normal Form is defined to disallow multivalued attributes, composite attributes and their combinations. It states that the domain of an attribute must include only atomic(single) values. There are three main techniques to achieve first normal form:

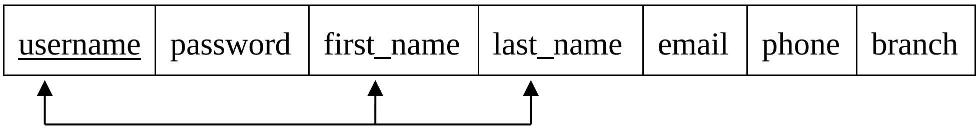
1. Removing the attribute which violates 1NF.
2. Expanding the key.
3. Expanding the multivalued attribute into ‘n’ atomic values. In Figure 2.1, USER had a multivalued attribute called name. It has been decomposed into another table having atomic attributes – first_name and last_name. Hence, in this project, all the tables are in First Normal Form.

USER



<u>username</u>	password	name	email	phone	branch

USER



<u>username</u>	password	first_name	last_name	email	phone	branch

3.4.2 SECOND NORMAL FORM (2NF)

It is based on the concept of full functional dependency. A functional dependency X→Y is a full functional dependency if removal of any attribute A from X means that the dependency does not hold anymore. In the figure 2.2, EVENT_DETAILS is normalized into EVENTS and QUESTIONS

The basic definition of 2NF: A relation schema R is in 2NF if every prime attribute A in R is fully functional dependent on the primary key of R.

The EVENT_DETAILS has all the attributes of events and also its related questions. If we want to use the combination of both the tables we can use JOIN operations and then we can get a table which has the combination of both events and its respective questions.

EVENT_DETAILS

event_id	event_name	description	organizers	venue	time	date	limit_max	fee	question_id
question	option1	option2	option3	option4					

EVENTS

event_id	event_name	description	organizers	venue	time	date	limit_max	fee

QUESTIONS

event_id	question_id	question	option1	option2	option3	option4

3.4.3 THIRD NORMAL FORM (3NF)

- It is based on the concept of transitive dependency.
- A functional dependency X→Y is a transitive dependency if there exists a set of attributes Z in R (relational schema) that is neither a candidate key nor a subset of any key of R and both X→Z and Z→Y holds.
- The basic definition of 3NF: A relation schema R is in 3NF if it satisfies 2NF and no non-prime attribute of R is transitively dependent on the primary key.
- In this project, we have no instance of a transitive dependency. Thus, this relational schema is in 3NF.

CHAPTER 4

IMPLEMENTATION

4.1 Front End Design

HTML code Snippet:

```
<div class="wrap-input100 validate-input m-b-26" data- validate="Username is
required"><span class="label-input100">Username</span>
<input class="input100" type="text" name="username" placeholder="Enter
username">
</div>
```

The above code is to take the username as an input in the form of text. The autocomplete provides predictions as the user name is being entered in the text box. The autofocus attribute is a Boolean attribute, it specifies that the element should automatically get focus when the page loads.

```
<div class="wrap-input100 validate-input m-b-18" data – validate = "Password is required”>
<span class="label-input100">Password</span>
<input class="input100" type="password" name="password" placeholder="Enter password">
```

The above code is to take the password as an input in the form of text. It only accepts if the user enters minimum of one character.

```
<button class=”login100-form-btn”>Login</button>
```

The above code is for login button. When the user clicks the button after filling the required details, the details are verified and then he/she is logged in.

4.2 Back End Design

Python code Snippet:

```
def user_login(request):
        if request.method ==’POST’:
                username = request.POST[‘username’]
                password = request.POST[‘password’]
```

}

```
                cursor = connection.cursor()
                cursor.execute(“SELECT password FROM USERS WHERE username = %s”,
                [username])
                context=cursor.fetchone()
                if password == context[0]:
                        request.session[11]=username
                        return redirect(‘home/’)
                else:
                        message.error(request,”Invalid username or password.”)
        return render(request=request,template_name = “Login.html/”)
```

The above code extracts username and password which the user enters in the front-end and then verifies that with the database if its correct or not and then logs in the user.

4.3 Database Connectivity

Django is a web application development framework written in the Python programming language. It is designed to make programming web applications easier by making assumptions about what every developer needs to get started. It allows you to write less code while accomplishing more than many other languages and frameworks. The django contains a python file called settings.py where in all these details of the database can be found.

Following is the code to configure the database:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'djangoproject',
        'USER' : 'gopal',
        'PASSWORD' : 'gopal123',
        'HOST' : 'localhost',
    }
}
```

4.4 Stored Procedures

A stored procedure is a set of Structured Query Language (SQL) statements with an assigned Name, which are stored in a relational database management system as a group, so it can be reused and shared by multiple programs. **“**ins_ques**”** This inserts the event data into the events table.

```
CREATE PROCEDURE ins_ques(character varying, integer, character varying, character
varying, character varying, character varying, character varying) LANGUAGE 'plpgsql'
AS $BODY$
        BEGIN
        INSERT INTO Question (event_id, question_id, question, option1, option2, option3,
        option4) VALUES ($1, $2, $3, $4, $5, $6, $7);
        COMMIT;
END;
$BODY$;
```

4.5 Triggers

A database trigger is procedural code that is automatically executed in response to certain events on a particular table or view in a database. The trigger is mostly used for maintaining the integrity of the information on the database.

```
        CREATE TRIGGER registered_on
        AFTER INSERT ON Registered
        BEGIN
        NEW.registered_on = CURRENT_DATE;
        return NEW;
        END;
```

This trigger inserts the current date to an event after the user registers for a given event. When the user registeres for an event , automatically the current date is fetched and put into the database.

4.6 Queries

These are few of the operation performed in the backend.

- SELECT * FROM events WHERE event_id=%s",[eventid]

  This query is used to select all the columns from the table 'Events' where event id is same as the passed variable event_id. This is used in the homepage of the admin and user. An user can select the event which he is interested in and the registeres to that particular event.

- DELETE FROM Registered WHERE user_id=%s AND event_id = %s",(user_id,event_id)

  This query is used to unregister a student with user id same as passed variable 'user_id' from the event where event id is same as the passed variable 'event_id'. This is used by the user whenever he wants to de-register from a particular event. He first selects the events and then clicks deregister. Hence this query is exected at that point.

- UPDATE user SET username=%s, email=%s, password=%s, first_name=%s, last_name=%s, phone=%s,branch=%s WHERE username=%s", (username, email, password, firstname, lastname, phone, branch, username)

  This query is used to update the student details in the table 'User' with the passed variables. This query is used to update the details of the user which can be one among the following: username, email, password, first_name, last_name, phone, branch by proving the username of the user which the admin wants to modify.

CHAPTER 5

TESTING

Sl. No	INPUT	OUTPUT	REMARKS
1	Open in chrome, pgAdmin	Localhost refused to connect Try checking connection ERR_CONNECT ION	pgAdmin not started i.e. postgreSQL server not started.
2	Open in chrome, the localhost	Localhost refused to connect Try checking connection ERR_CONNECT I ON	Python 18development Server not started.
3	Insertion insert into users values (’1BI17IS037','pass,'Gopal',’Krishna’,gopal @gmail.com','7878787878',’ISE’')	ERROR Username already taken	UNIQUE constrain t violated. Integrity constraint.
4	Insertion insert into users values (’1BI17IS034','pass,'Rajat',’Jain’,raju@gm ai l.com','7878787878',’ISE’')	Successfully added to the database	Success

Table 5.1 Testing

CHAPTER 6

SNAPSHOTS

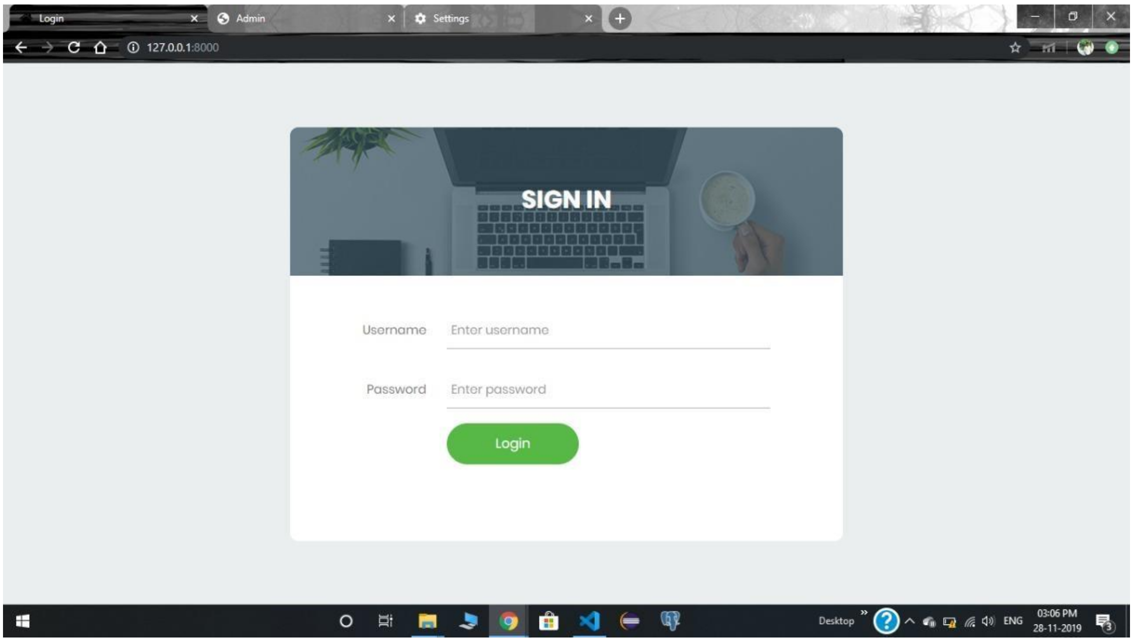


Figure 6.1 Login Page

The user needs to enter the username and password to login

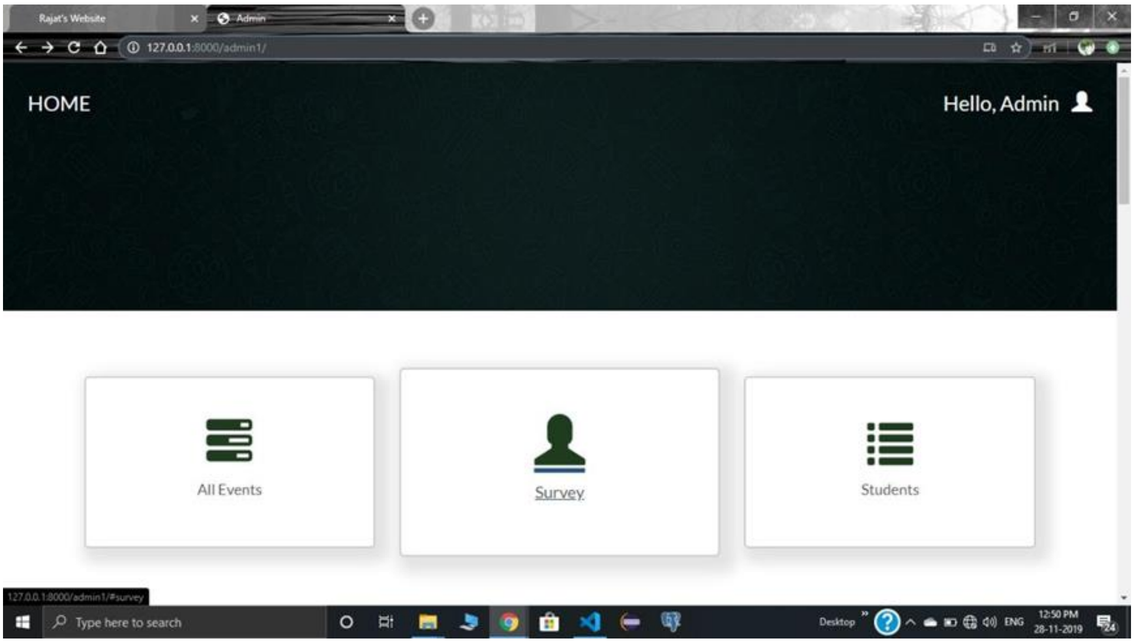
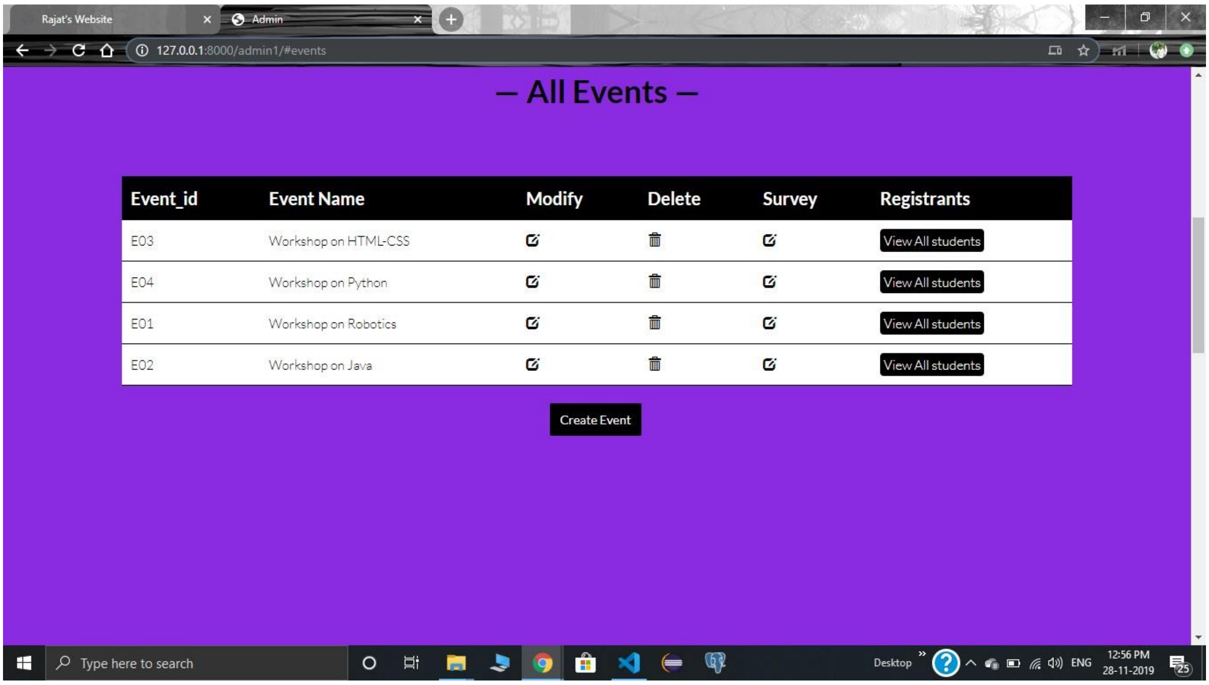


Figure 6.2 Admin Panel

This page gives admin to choose from provided options



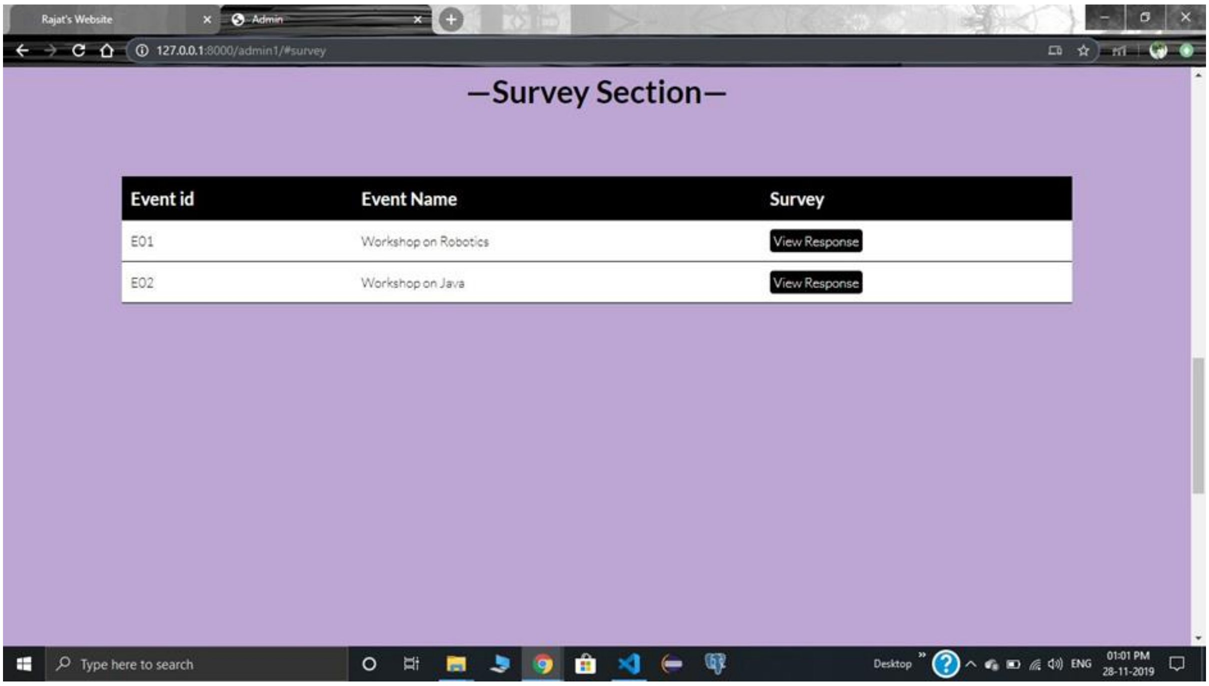
— All Events —

Event_id	Event Name	Modify	Delete	Survey	Registrants
E03	Workshop on HTML-CSS	✎	🗑	✎	View All students
E04	Workshop on Python	✎	🗑	✎	View All students
E01	Workshop on Robotics	✎	🗑	✎	View All students
E02	Workshop on Java	✎	🗑	✎	View All students

Create Event

Figure 6.3 All Events

This page shows all the events created by the admin.

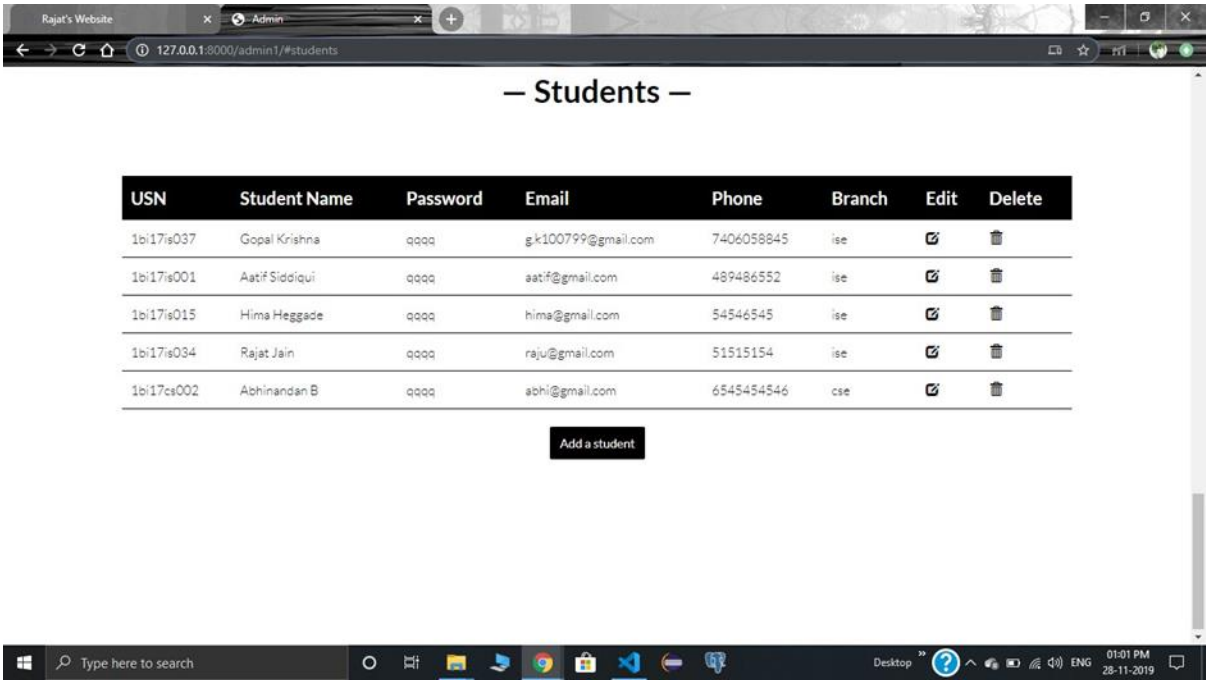


—Survey Section—

Event id	Event Name	Survey
E01	Workshop on Robotics	View Response
E02	Workshop on Java	View Response

Figure 6.4 Survey Section

This page shows the completed events with an option to view it's survey response.

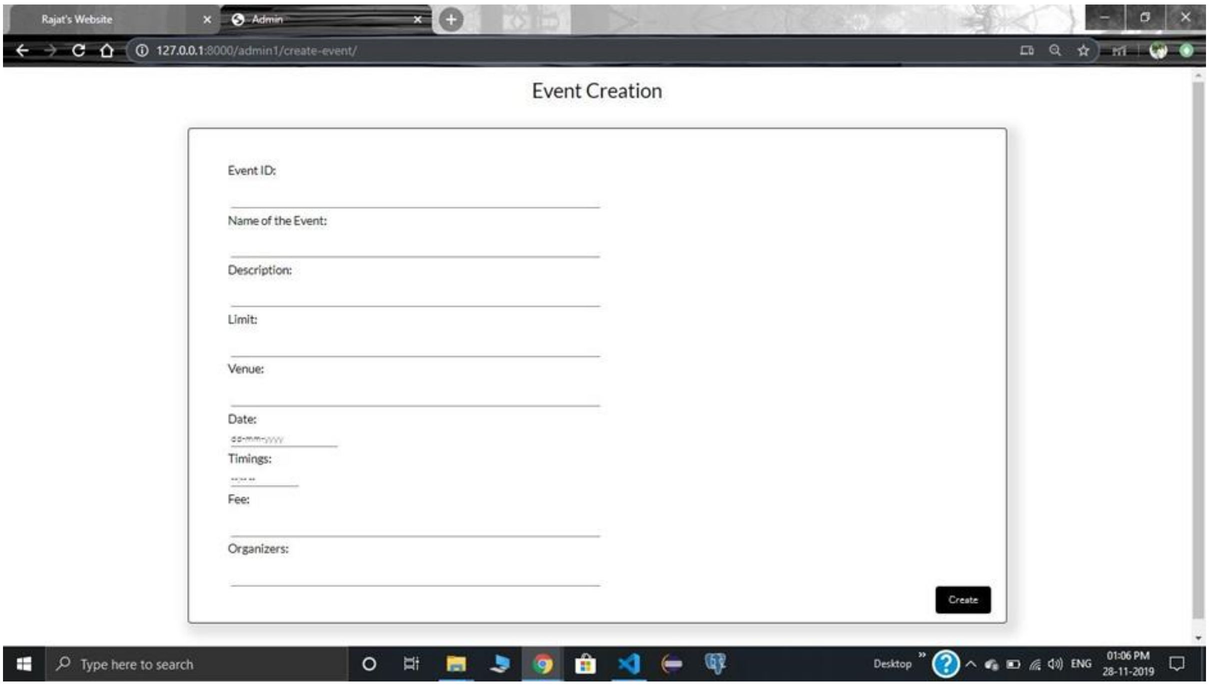


— Students —

USN	Student Name	Password	Email	Phone	Branch	Edit	Delete
1bi17is037	Gopal Krishna	qqqq	g.k100799@gmail.com	7406058845	ise	✎	🗑
1bi17is001	Aatif Siddiqui	qqqq	aatif@gmail.com	489486552	ise	✎	🗑
1bi17is015	Hima Heggade	qqqq	hima@gmail.com	54546545	ise	✎	🗑
1bi17is034	Rajat Jain	qqqq	raju@gmail.com	51515154	ise	✎	🗑
1bi17cs002	Abhinandan B	qqqq	abhi@gmail.com	6545454546	cse	✎	🗑

Figure 6.5 Students Section

The admin can view details of all the students from the database.



Event Creation

Event ID:

Name of the Event:

Description:

Limit:

Venue:

Date:
dd-mm-yyyy

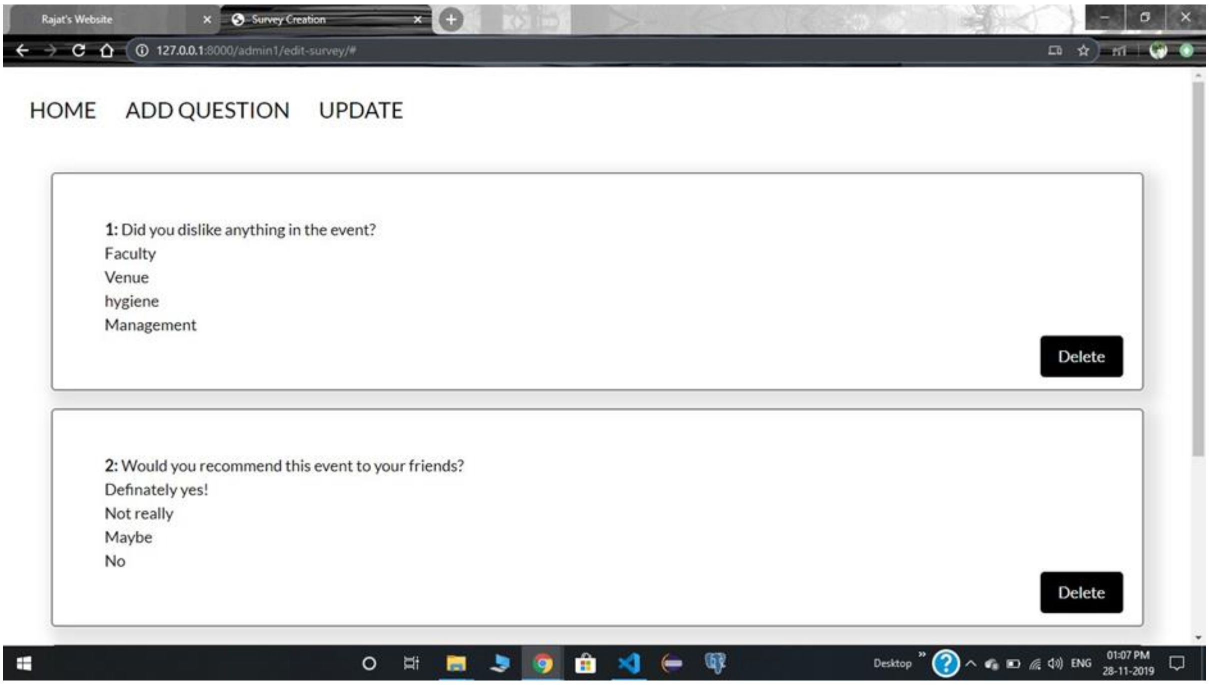
Timings:
--:-- --

Fee:

Organizers:

Figure 6.6 Event Creation by admin

Admin can create events through this page.

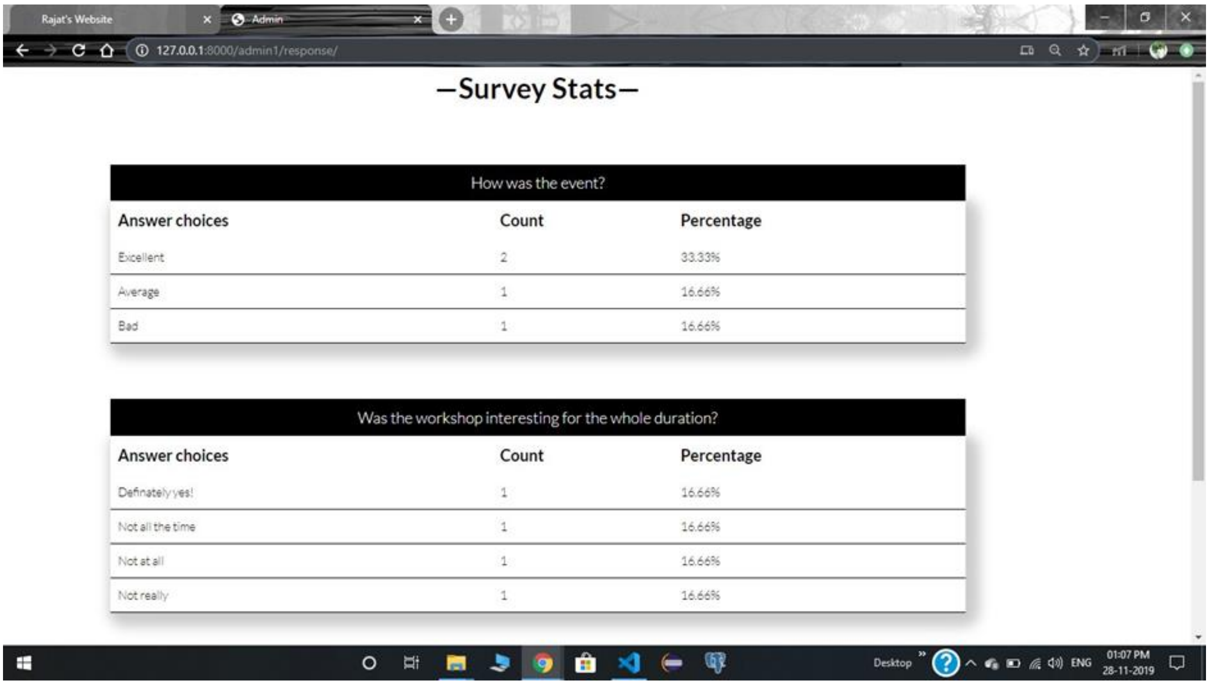


1: Did you dislike anything in the event?
Faculty
Venue
hygiene
Management

2: Would you recommend this event to your friends?
Definately yes!
Not really
Maybe
No

Figure 6.7 Survey Creation

The admin can create questions in the survey form provided to students.



—Survey Stats—

How was the event?

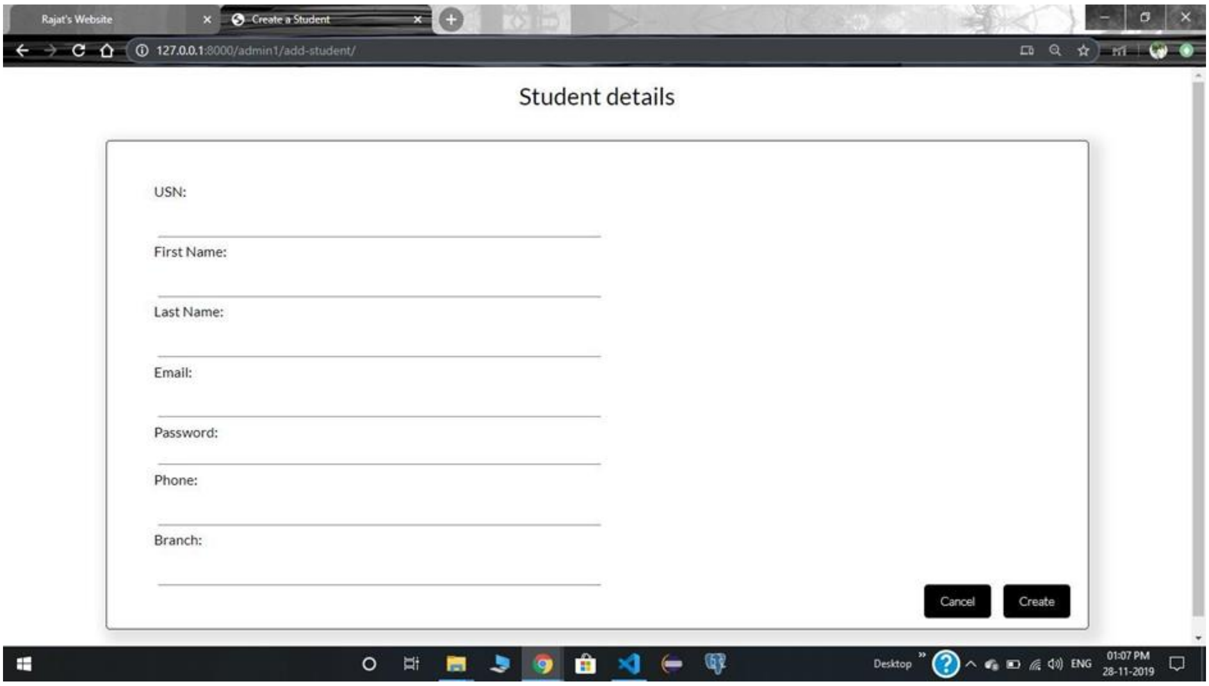
Answer choices	Count	Percentage
Excellent	2	33.33%
Average	1	16.66%
Bad	1	16.66%

Was the workshop interesting for the whole duration?

Answer choices	Count	Percentage
Definately yes!	1	16.66%
Not all the time	1	16.66%
Not at all	1	16.66%
Not really	1	16.66%

Figure 6.8 Survey Stats

The admin can view student's responses to the events organized.



Student details

USN:

First Name:

Last Name:

Email:

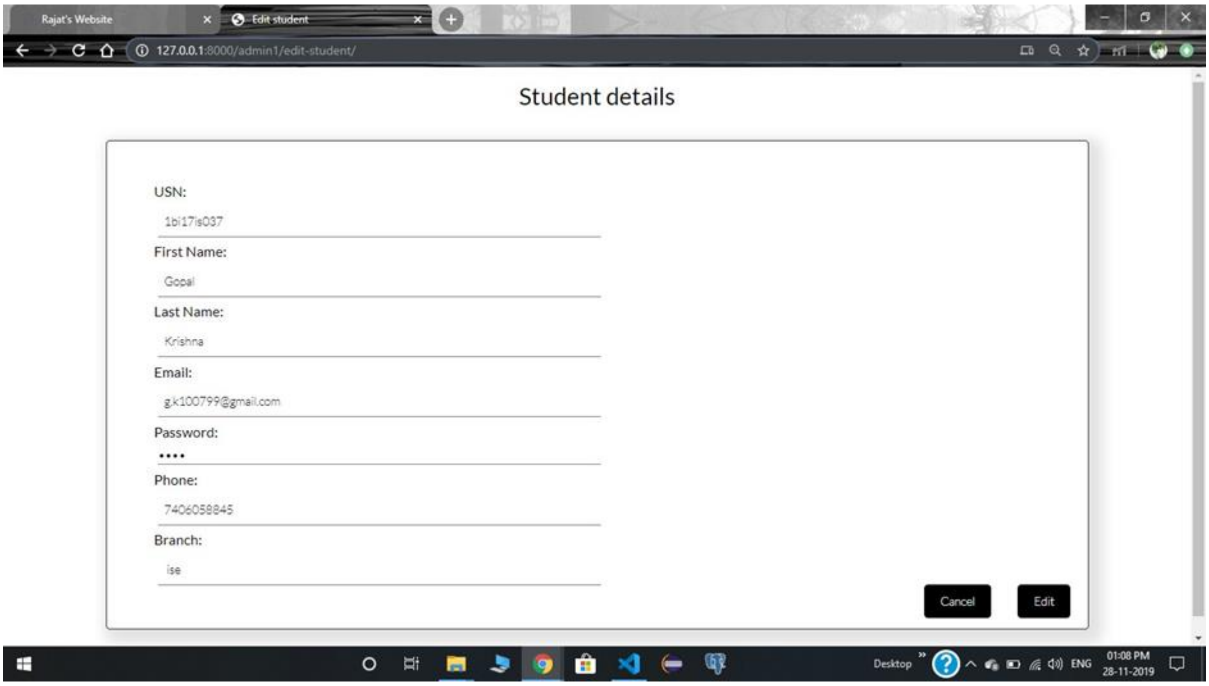
Password:

Phone:

Branch:

Figure 6.9 Create Student Form

The admin can create a student entry to the database through this form.



Student details

USN: 1bi17is037

First Name: Gopal

Last Name: Krishna

Email: g.k100799@gmail.com

Password: ••••

Phone: 7406058845

Branch: ise

Figure 6.10 Editing the student details

The admin can edit the student details through this form.

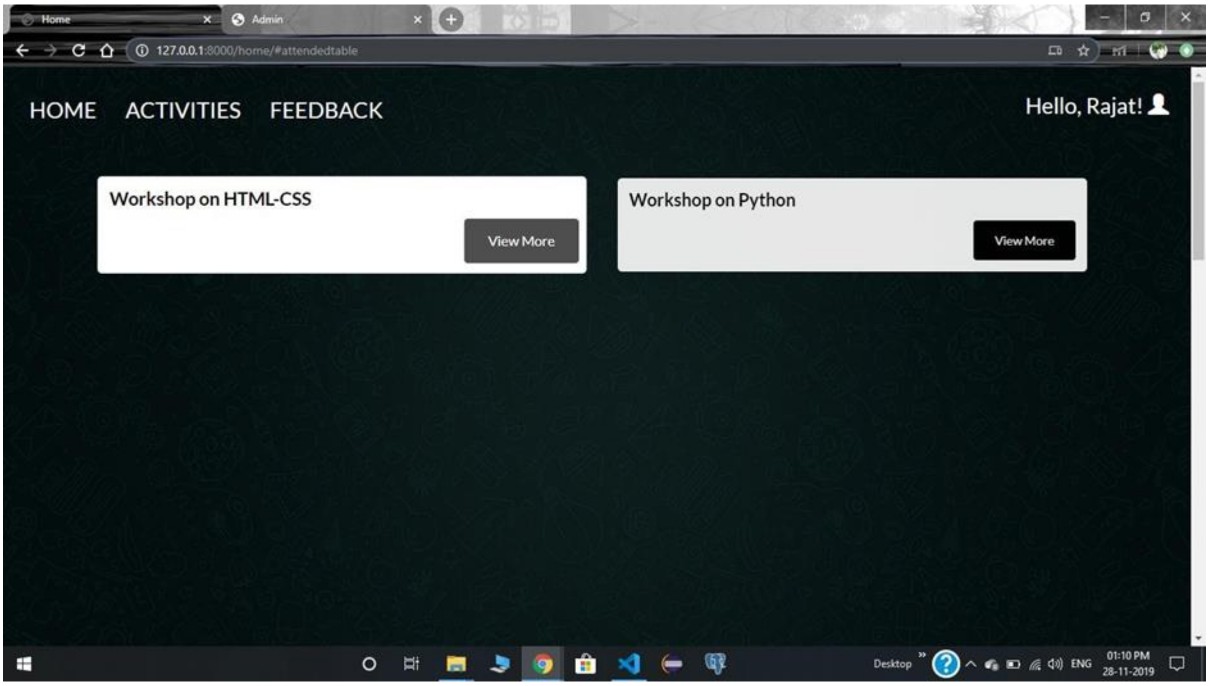
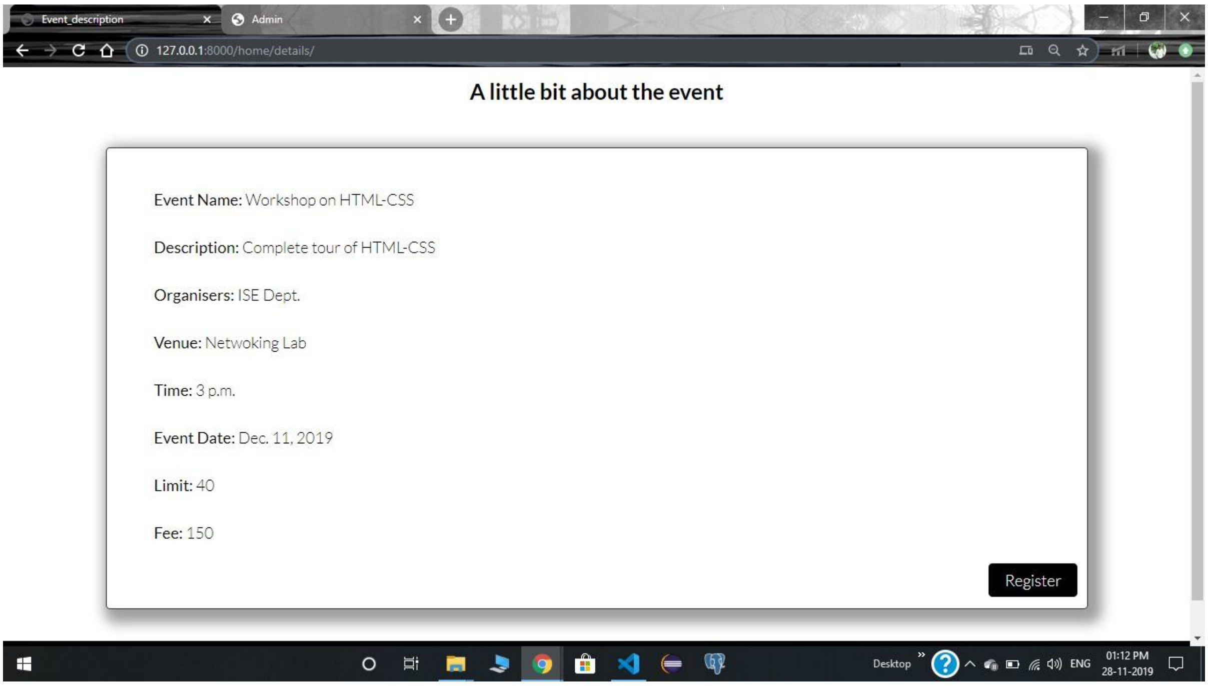


Figure 6.11 Student Homepage

Once the user completes the examination, the result is instantly displayed.



A little bit about the event

Event Name: Workshop on HTML-CSS

Description: Complete tour of HTML-CSS

Organisers: ISE Dept.

Venue: Netwoking Lab

Time: 3 p.m.

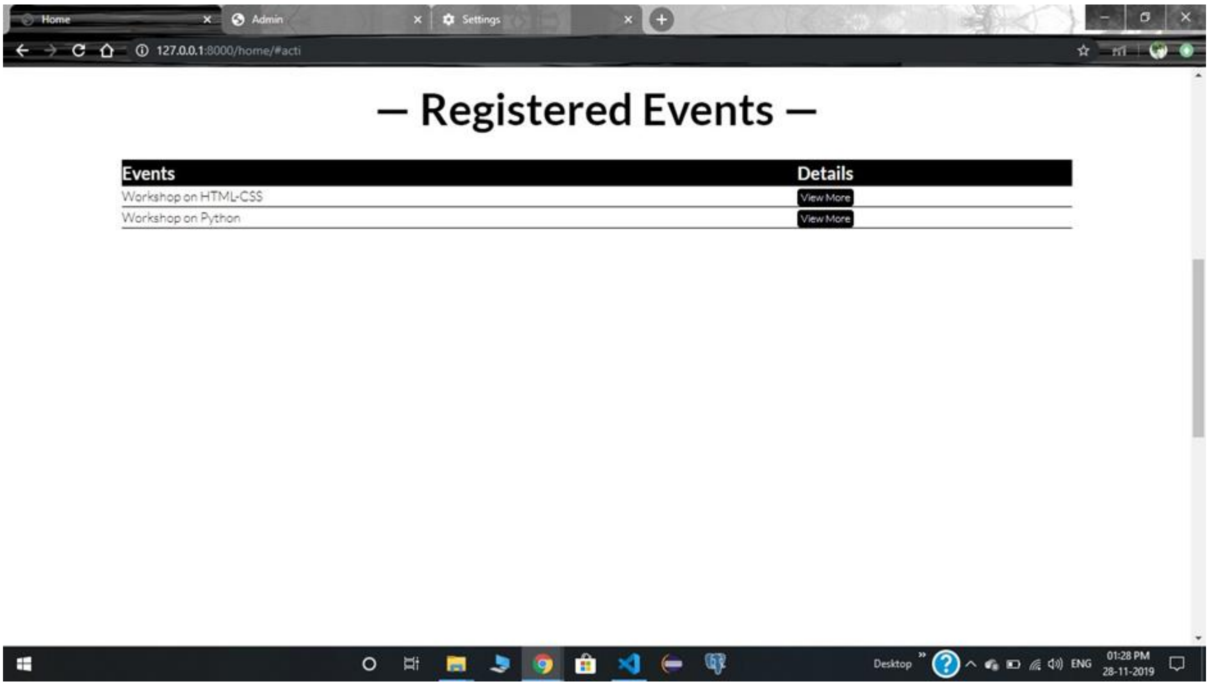
Event Date: Dec. 11, 2019

Limit: 40

Fee: 150

Figure 6.12 Event Description

This page gives user the information about the events available.

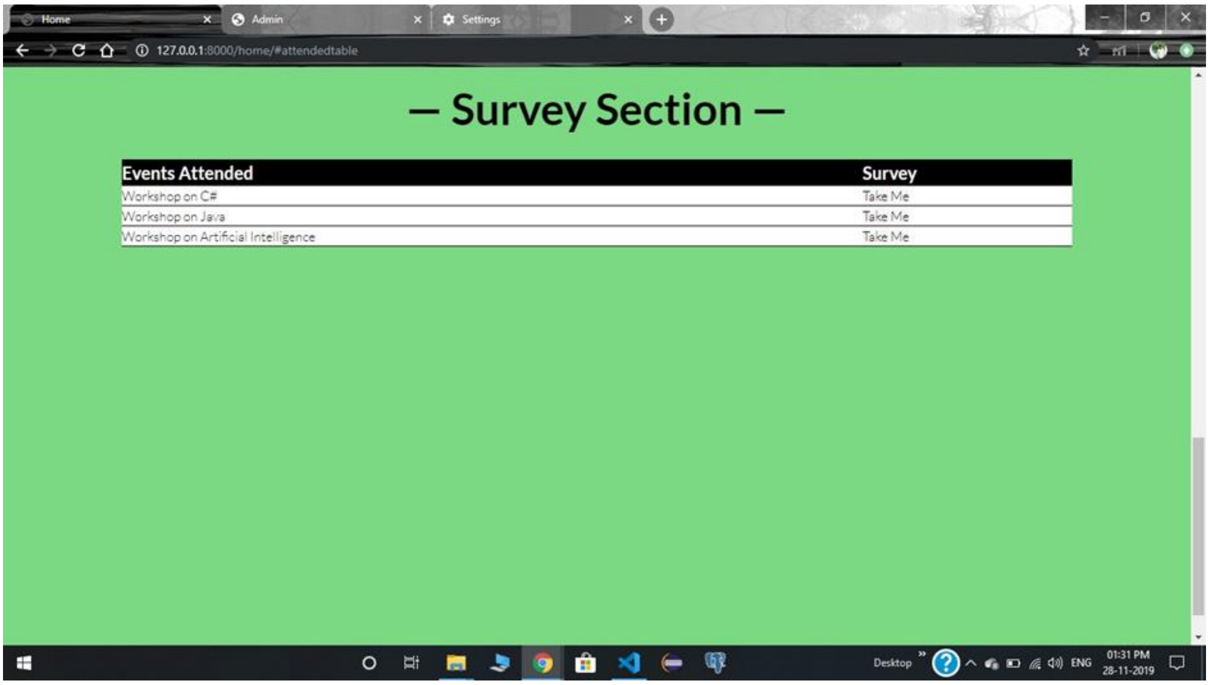


— Registered Events —

Events	Details
Workshop on HTML-CSS	View More
Workshop on Python	View More

Figure 6.13 Events Registered

This page shows all the events that the user has registered.



— Survey Section —

Events Attended	Survey
Workshop on C#	Take Me
Workshop on Java	Take Me
Workshop on Artificial Intelligence	Take Me

Figure 6.14 Survey Section

This page lists the events for which the user can give survey.

CHAPTER 7

CONCLUSION

From proper analysis of positive points and constraints on component, it can be concluded that the product is a highly efficient and will be of great help to the general public. This can be run on any system. The website is user friendly and is available to all. Manual method can be completely wiped off due to introduction of the webpage

FUTURE ENHANCEMENT

- The system is just a simple database design but still designed at maximum possible excellence.
- Still I accept drawbacks, as it is a human effort.
- Website security must be increased.
- Some complex queries to acquire multiple table data to retrieve can be included also, delete column can be provided to delete entries.
- There can be many more future enhancement and improvement in the database system designed.

REFERENCES

[1]. www.w3schools.com

[2]. www.stackoverflow.com

[3]. www.tutorialspoint.com

[4]. Elmasri and Navathe : Fundamental Database Management,7th edition 2017