

**“RJ EVENTS”**  
**BACHELOR OF TECHNOLOGY**  
in  
**COMPUTER SCIENCE AND ENGINEERING**



**RGUKT**

**Rajiv Gandhi University of Knowledge Technologies**

**R.K.VALLEY**

Submitted by

E Jahnavi – R170415

K Reshma – R170427

Under the Esteemed guidance of

Mr.Satya Nandaram N

RGUKT RK Valley.

## **DECLARATION**

We hereby declare that the report of the B.Tech Major Project Work entitled “RJ EVENTS” which is being submitted to Rajiv Gandhi University of Knowledge Technologies, RK Valley, in partial fulfillment of the requirements for the award of Degree of Bachelor of Technology in Computer Science and Engineering, is a bonafide report of the work carried out by us. The material contained in this report has not been submitted to any university or institution for award of any degree.

K Reshma -- R170427

E Jahnavi -- R170415

Dept. Of Computer Science and Engineering

# **RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES**



**RGUKT**

(A.P. Government Act 18 of 2008)  
RGUKT, RK VALLEY

Department of Computer Science and Engineering

## **CERTIFICATE FOR PROJECT COMPLETION**

This is to certify that project entitled “RJ EVENTS” submitted by E Jahnavi (R170415), K Reshma (R170427), under our guidance and supervision for the partial fulfillment for the degree Bachelor of Technology in Computer Science and Engineering during the academic semester 2 2021-2022 at RGUKT, RK VALLEY. TO the best of my knowledge, the results embodied in this dissertation work have not been submitted to any University or Institute for the award of any degree or diploma.

### **Project Internal Guide**

Mr. N. Satya Nandaram

Assistant Professor

RGUKT, RK Valley

### **Head of the Department**

Mr. P. Harinadha

HOD Of CSE

RGUKT, RK Valley

## **Index**

1. Introduction -----	5
2. Abstraction -----	6
3. Objectives -----	7
4. Intended Audience -----	7
5. Product Vision -----	7
6. Scope -----	8
7. Modules -----	9
8. Technologies -----	9
8.1 HTML -----	9
8.2 CSS -----	9
8.3 JAVA SCRIPT -----	10
8.4 PYTHON -----	10
8.5 DJANGO -----	10
8.6 DB SQLite -----	10-11
9. System in context -----	12-14
10. Starting Project -----	15-16
11. Functional Requirements -----	17
12. Admin Page -----	18-20
13. Source Code -----	21-24
14. Web Pages -----	25-31
15. References-----	31

## **1.Introduction**

The “Event Management System” has been developed to override the problems prevailing in the practicing manual system. This software 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 company to carry out operations in a smooth and effective manner.

The application is reduced as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is needed for the user to use this system. Thus by this all it proves it is user-friendly. Event Management System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus it will help organization in better utilization of resources.

Every organization, whether big or small, has challenges to overcome and managing the information of Activity, Event, Attendees, Payment, Conductors. Every Event Management System has different Event needs, therefore we design exclusive employee management systems that are adapted to your managerial requirements. This is designed to assist in strategic planning, and will help you ensure that your organization is equipped with the right level of information and details for your future goals. Also, for those busy executive who are always on the go, our systems come with remote access features, which will allow you to manage your workforce anytime, at all times. These systems will ultimately allow you to better manage resources.

## **2. Abstract of the Project Event Management System:**

The purpose of Event management System is to automate the existing manual system by the help of computerized equipments and full-fledged computer software, 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.

Event Management System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus it will help organization in better utilization of resources. The organization can maintain computerized records without redundant entries. That means that one need not be distracted by information that is not relevant, while being able to reach the information.

The aim is to automate its existing manual system by the help of computerized equipments and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. Basically the project describes how to manage for good performance and better services for the clients.

### **3.Objective of Project on Event Management System:**

The main objective of the project on Event Management System is to manage the details of Event, Activity, Payment, Decoration, Organizers. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the Event, Activity, Payment, Decoration, Organizers. It tracks all the details about the Organizers and their requirements for an Event.

### **4.Intended Audience:**

The intended audience will be the users who can sign up to access the Event Management System and also users can check the events and register for the event which they want.

### **5.Product Vision:**

#### **Vision Statement:**

The product vision is to develop an Event Management, which is a user friendly and easily accessible. This RJ Events helps to provide services to user which will decrease manual work for him and make event successfully.

## **6.Scope of the project Event Management System:**

It may help collecting perfect management in details. In a very short time, the collection will be obvious, simple and sensible. It will help a person to know the management of passed year perfectly and vividly. It also helps in current all works relative to Event Management System. It will be also reduced the cost of collecting the management & collection procedure will go on smoothly.

- Our project aims at Business process automation, i.e., we have tried to computerize various processes of Event Management System.
- In computer system, the person has to fill the various forms & number of copies of the forms can be easily generated at a time.
- To assist the staff in capturing the effort spent on their respective working areas.
- To utilize resources in an efficient manner by increasing their productivity through automation.
- The system generates types of information that can be used for various purposes.
- It satisfy the user requirement.
- Be easy to understand by the user and operator.
- Be easy to operate.
- Have a good user interface.
- Be expandable.
- Delivered on schedule within budget.



## **7.Modules of Event Management System:**

- Events Module: Used for managing the Event details.
- Sign Up Module: Used for managing the Sign Up details.
- Login Module: Used for managing the Login details.
- Home Module: Used for managing the Web Page details.
- About Us Module: Used for managing the Managers details.
- Registration Module: Used for managing the Registration details.

## **8.Technologies:**

- Html
- CSS
- Java Script
- Python
- Django
- DB SQLite

### **8.1 Html:**

HTML is a Hyper Text Markup Language used to create electronic documents (called pages) that are displayed on the World Wide Web. Each page contains several connections to other pages called hyperlinks. Every web page you see was written using one version of HTML.

### **8.2 CSS:**

Cascading Style Sheets (CSS) is a styleshet language used to describe the presentation of a document written in HTML or XML. CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

### **8.3 JavaScript:**

JavaScript is a scripting language that enables you to create dynamically updating content, control multimedia, animate images, and pretty much everything else.

### **8.4: Python:**

Python is a very popular general-purpose interpreted, interactive, object-oriented, and high-level programming language. Python is dynamically-typed and garbage-collected programming language. It was created by Guido van Rossum during 1985- 1990.

### **8.5: Django:**

Django is a high-level Python web framework tht enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel.

### **8.6: DB SQLite:**

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine.

SQLite is used to develop embedded software for devices like televisions, cell phones, cameras etc. It can manage low to medium-traffic HTTP requests. SQLite can change files into smaller size archives with lesser metadata. SQLite is used as a temporary dataset to get processed with some data within an application.

## Connection of DBSQLite in Django :

The screenshot shows the Visual Studio Code interface with the `settings.py` file open. The `DATABASES` section is configured to use SQLite. The `DEBUG` flag is set to `True`. The `ALLOWED_HOSTS` list is empty. The `INSTALLED_APPS` list includes `django.contrib.auth`, `django.contrib.contenttypes`, `django.contrib.sessions`, `django.contrib.messages`, `django.contrib.staticfiles`, and `django.contrib.admin`. The `TEMPLATES` list includes `django.template.backends.django.DjangoTemplates`. The `STATIC_URL` is set to `'/static/'`. The `STATICFILES_DIRS` list includes `os.path.join(BASE_DIR, 'static')`. The `SECRET_KEY` is set to `'django-insecure-...'`. The `DEFAULT_AUTO_FIELD` is set to `django.db.models.AutoField`. The `EMAIL_BACKEND` is set to `django.core.mail.backends.console.EmailBackend`. The `LOGGING` configuration is also present.

```
13 # Django settings for project.
14
15 # Build paths inside the project.
16 BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
17
18 # Quick-start development settings - unsuitable for production
19 # See https://docs.djangoproject.com/en/1.11/ref/settings/#databases
20
21 # Database
22 # https://docs.djangoproject.com/en/1.11/ref/settings/#databases
23
24 DATABASES = {
25     'default': {
26         'ENGINE': 'django.db.backends.sqlite3',
27         'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
28     }
29 }
30
31 # Password validation
32 # https://docs.djangoproject.com/en/1.11/ref/settings/#auth-password-validators
33
34 AUTH_PASSWORD_VALIDATORS = [
35     {
36         'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
37     },
38 ]
39
40 # Internationalization
41 # https://docs.djangoproject.com/en/1.11/topics/i18n/
42 LANGUAGE_CODE = 'en-us'
43 TIME_ZONE = 'UTC'
44 USE_I18N = True
45 USE_TZ = True
46
47 # Static files (CSS, JavaScript, Images)
48 # https://docs.djangoproject.com/en/1.11/topics/static/
49 STATIC_URL = '/static/'
50
51 # Media files (images, pdf, videos)
52 # https://docs.djangoproject.com/en/1.11/topics/media/
53 MEDIA_URL = '/media/'
54
55 # Default primary key field type
56 # https://docs.djangoproject.com/en/1.11/ref/settings/#default-auto-field
57 DEFAULT_AUTO_FIELD = django.db.models.AutoField
58
59 # Email settings
60 EMAIL_BACKEND = 'django.core.mail.backends.console.EmailBackend'
61
62 # Logging
63 LOGGING = {
64     'version': 1,
65     'disable_existing_loggers': False,
66     'handlers': {
67         'console': {
68             'class': 'logging.StreamHandler',
69         },
70     },
71     'loggers': {
72         'django': {
73             'handlers': ['console'],
74             'level': INFO,
75             'propagate': True,
76         },
77     },
78 }
```

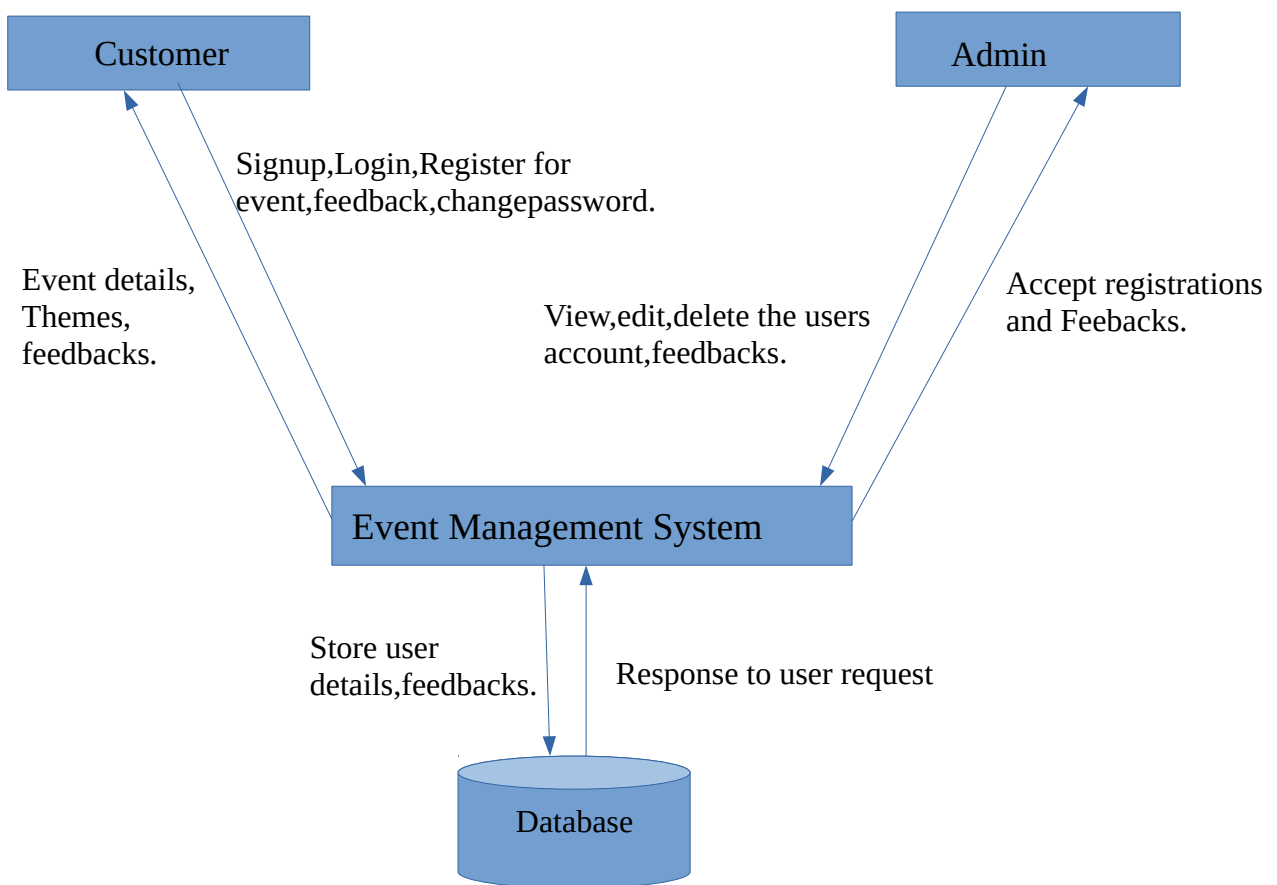
The screenshot shows the Visual Studio Code interface with the `db.sqlite3` database open. The `auth_user` table is selected, showing 12 records. The table has columns: `id`, `password`, `last_login`, `is_superuser`, `first_name`, `last_name`, and `email`. The records show users with various passwords and last login times.

id	password	last_login	is_superuser	first_name	last_name	email
1	pbkdf2_sha256\$3600\$...	2022-09-20 02:16:01...	1	Jahnavi	Ekoti	ekotijahnavi050...
2	pbkdf2_sha256\$3600\$...	2022-09-19 05:31:27...	0	Reshma	Kalluru	reshmakalluru8...
3	pbkdf2_sha256\$3600\$...	2022-09-20 02:21:52...	0	Geethanjali	K	geethu@gmail.c...
4	pbkdf2_sha256\$3600\$...	2022-09-19 06:34:40...	0	Ramani	Medhuru	ramani@gmail.c...
5	pbkdf2_sha256\$3600\$...	NULL	0	Jayasree	Ekoti	ekotijayasree@...
6	pbkdf2_sha256\$3600\$...	2022-09-19 11:27:41...	0	Nagamuneiah	Kalluru	nanikalluru600...
7	pbkdf2_sha256\$3600\$...	NULL	0	kaliya	chota	kaliya@gmail.co...
8	pbkdf2_sha256\$3600\$...	NULL	0			kaliyabh@gmail...
9	pbkdf2_sha256\$3600\$...	NULL	0	Roshni	Kalluru	roshnikalluru60...
10	pbkdf2_sha256\$3600\$...	NULL	0	Falak	Shahin	falakshahin@gr...

## **9.System in Context:**

This RJ EVENTS provide the list of events and also provides the previous events models and user experience which already registered for an event.

### **Context Diagram:**



## **System-wide Requirements(Received):**

### **Actors:**

The system interacts with two kinds of users. Each user has its own functions to access system. The functionalities of users are dependent on each other.

## Events:

RJ Events is a multi-user system which provides activities associated with its day to day questions.

### The most critical events are:

1. Customer gets Signed up first using his name, email, username and password.
2. Customer login using the username and password.
3. Customer has to select an event.
4. After selecting event, he has to register for specific event.
5. In registration form, he has to give details of specific event and budget.
6. After that he has to submit registration form.

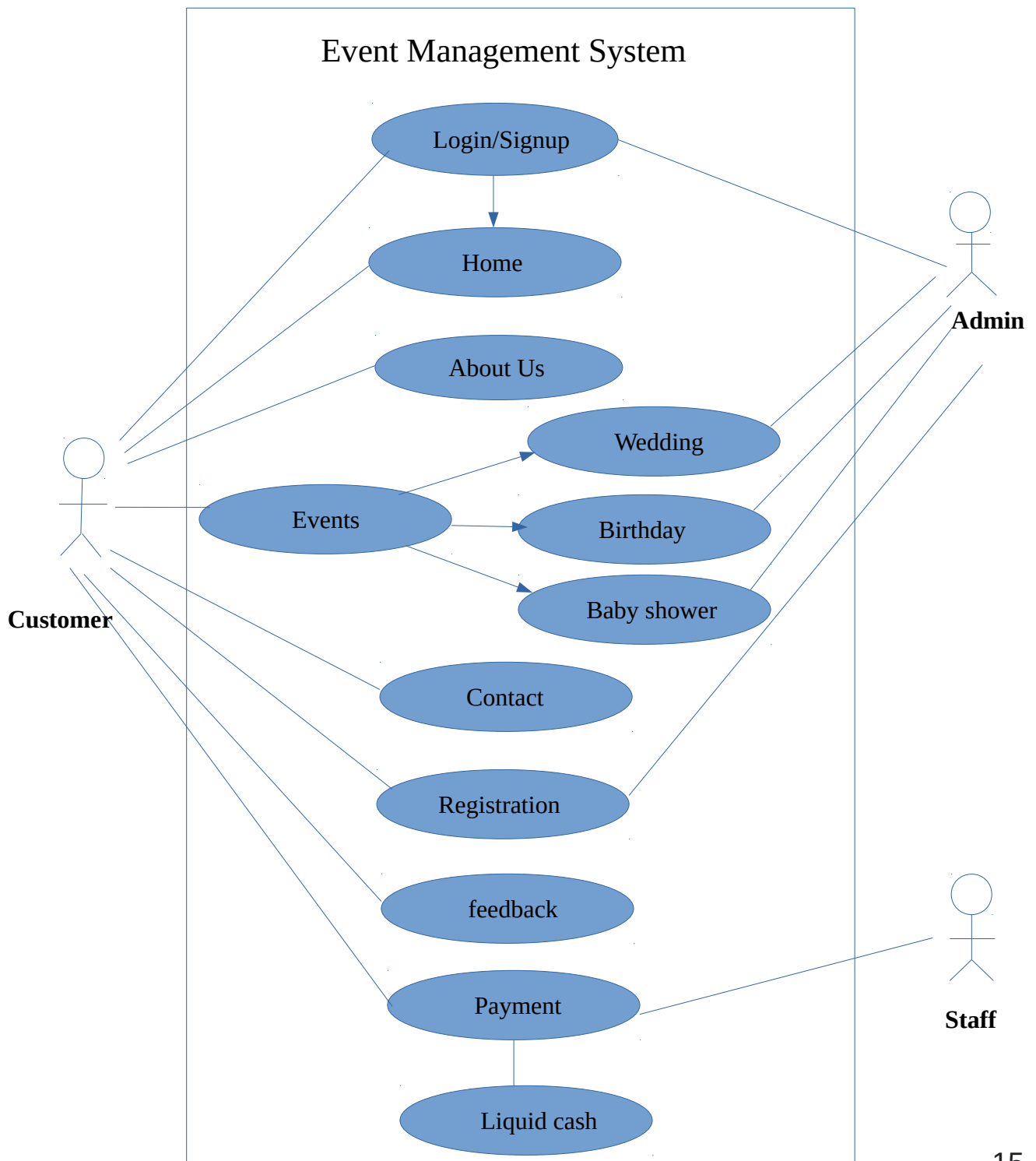
The below table provides a set of user visible events that define the functionalities that are in RJ Events.

	Actor	Action	Object	Frequency	Arrival Pattern	Response
1.	Customer	onclick	Signup	1/day	Episodic	It asks username first name last name email password
2.	Customer	onclick	Login	1/day	Episodic	It asks username password
3.	Customer	onclick	Main	1/day	Episodic	It displays all modules in the software
4.	Customer	onclick	Home	1/day	Episodic	It displays information about company
5.	Customer	onclick	About us	1/day	Episodic	It displays

						information about managers
6.	Customer	onclick	Events	1/day	Episodic	It displays all types of events
7.	Customer	onclick	Wedding	1/day	Episodic	It displays info about wedding and registration form
8.	Customer	onclick	Birthday	1/day	Episodic	It displays info about birthday and registration form
9.	Customer	onclick	Baby shower	1/day	Episodic	It displays information about baby shower functions and registration form
10.	Customer	onclick	Registration	1/day	Episodic	It displays event details and requirements for registering an event.

## 10.Functional Requirements:

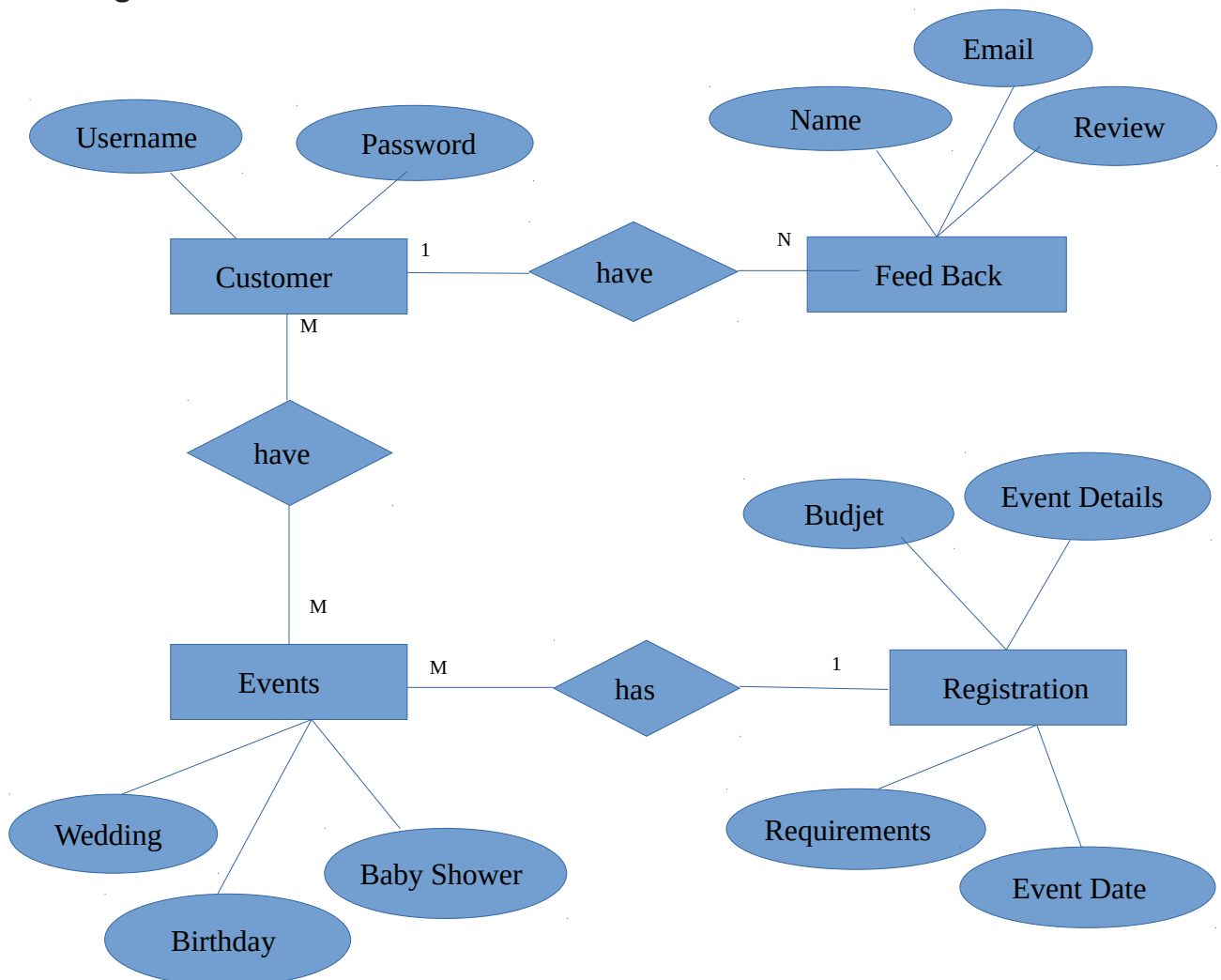
### Use Case Diagram:



### Use Case Overview:

	Use-case ID	Use-case Name	Priority	Stability	Verifiable
1.	UC-RJ-Ev	Events	High	Stable	Verifiable
2.	UC-RJ-RG	Registrations	High	Not Stable	Verifiable
3.	UC-RJ-PM	Previous Models	High	Stable	Verifiable
4.	UC-RJ-FB	FeedBack	High	Stable	Verifiable

### ER Diagram:





## **11. Starting Project:**

### **Steps:**

1.Install Python in the system

```
python --version
```

2.Install django by using pip

```
pip install django
```

3.Once we installed django in our system,we will get 'django-admin' command line tool ,which can be used to create our Django project. Create a project using : **django-admin startproject projectName**

By default some of the python files will be installed and database dbsqlite too. Open the project folder using visual studio code.

4.Create an application in that project folder

```
python3 manage.py startapp appname
```

5. Add application to the settings.py

6.Define view function inside views.py and define url-pattern to views inside urls.py

7. To Start Server run this following command in terminal

```
python3 manage.py runserver
```

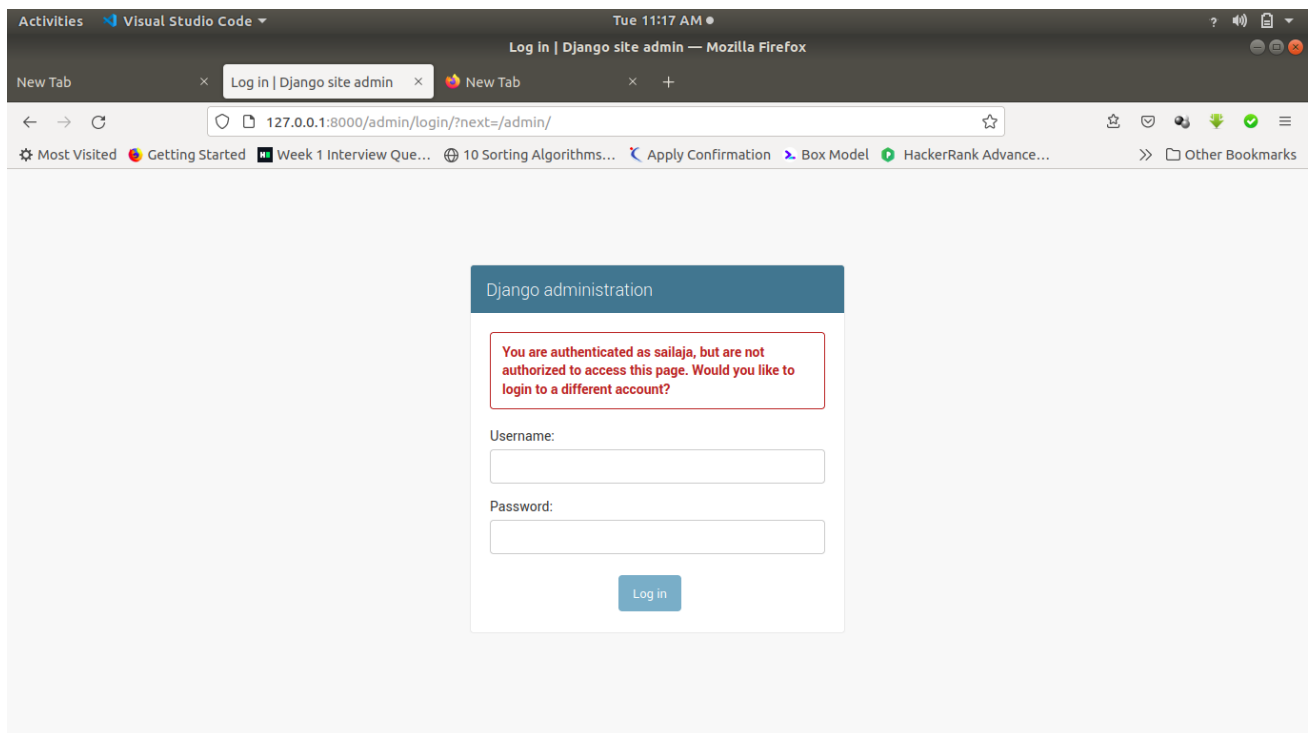
Make sure that run this command in the particular directory where this manage.py is present. This command will generate an url <http://127.0.0.1:8000/> use ctrl+click on the url to open in the browser.

8.Send the request.

## 12. Admin Page:

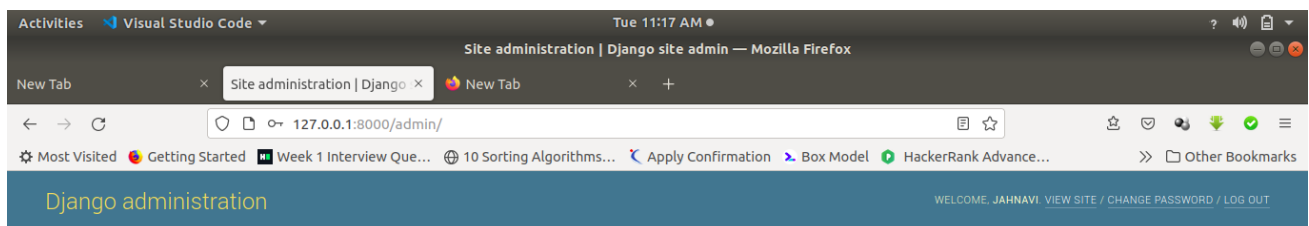
Django provides a default admin interface which can be used to perform create, read, update and delete operations on the model directly. It reads set of data that explain and gives information about data from the model, to provide an instant interface where the user can adjust contents of the application.

### Admin Login Page



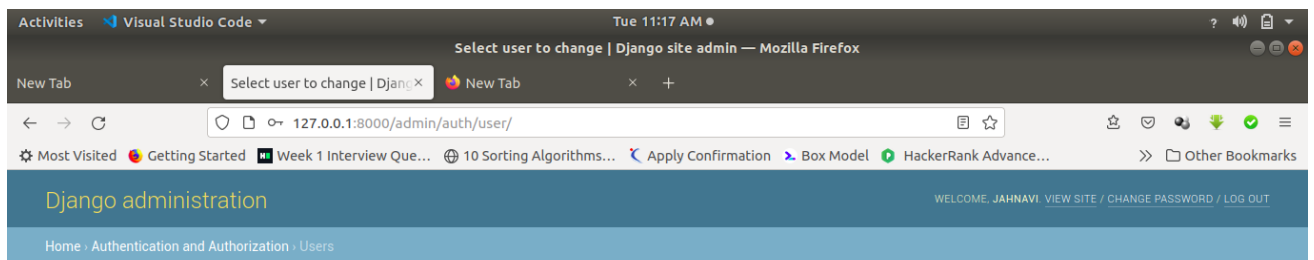
### Super User Creation:

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE
● student@r170415:~/project/project$ python3 manage.py createsuperuser
Username (leave blank to use 'student'): Reshma
Email address: reshmakalluru8680@gmail.com
Password:
Password (again):
The password is too similar to the email address.
Password:
Password (again):
Superuser created successfully.
○ student@r170415:~/project/project$
```



## Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
Users	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
MAIN		
Feed backs	<a href="#">+ Add</a>	<a href="#">✎ Change</a>
User requirementss	<a href="#">+ Add</a>	<a href="#">✎ Change</a>



Select user to change

<div> <input type="text"/> <input type="button" value="Search"/> </div>					
Action: <span>----- ▾</span> <input type="button" value="Go"/> 0 of 12 selected					
<input type="checkbox"/>	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
<input type="checkbox"/>	12345	kaliyabh@gmail.com	bheem	chota	<span>✖</span>
<input type="checkbox"/>	Falak	falakshahin@gmail.com	Falak	Shahin	<span>✖</span>
<input type="checkbox"/>	Jahnavi	ekotijahnavi0504@gmail.com	Jahnavi	Ekoti	<span>✔</span>
<input type="checkbox"/>	Jayasree	ekotijayasree@gmail.com	Jayasree	Ekoti	<span>✖</span>
<input type="checkbox"/>	Naga	nanikalluru600@gmail.com	Nagamuneiah	Kalluru	<span>✖</span>
<input type="checkbox"/>	Ramani	ramani@gmail.com	Ramani	Medhuru	<span>✖</span>
<input type="checkbox"/>	Reshma123	reshmakalluru8680@gmail.com	Reshma	Kalluru	<span>✖</span>
<input type="checkbox"/>	RoshniK	roshnikalluru600@gmail.com	Roshni	Kalluru	<span>✖</span>
<input type="checkbox"/>	amma	naqu@gmail.com	naqarani	ekoti	<span>✖</span>

Activities Visual Studio Code Tue 11:18 AM Select feed back to change | Django site admin — Mozilla Firefox

New Tab Select feed back to change New Tab

127.0.0.1:8000/admin/main/feedback/

Most Visited Getting Started Week 1 Interview Que... 10 Sorting Algorithms... Apply Confirmation Box Model HackerRank Advance... Other Bookmarks

## Django administration

WELCOME, JAHNAVI [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home Main Feed backs

### Select feed back to change

[ADD FEED BACK +](#)

Action:  Go 0 of 7 selected

<input type="checkbox"/>	FEED BACK
<input type="checkbox"/>	FeedBack object
<input type="checkbox"/>	FeedBack object
<input type="checkbox"/>	FeedBack object
<input type="checkbox"/>	FeedBack object
<input type="checkbox"/>	FeedBack object
<input type="checkbox"/>	FeedBack object
<input type="checkbox"/>	FeedBack object

7 feed backs

127.0.0.1:8000/admin/main/feedback/14/change/

Activities Visual Studio Code Tue 11:18 AM Select user requirements to change | Django site admin — Mozilla Firefox

New Tab Select user requirements to New Tab

127.0.0.1:8000/admin/main/userrequirements/

Most Visited Getting Started Week 1 Interview Que... 10 Sorting Algorithms... Apply Confirmation Box Model HackerRank Advance... Other Bookmarks

## Django administration

WELCOME, JAHNAVI [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home Main User requirements

### Select user requirements to change

[ADD USER REQUIREMENTS +](#)

Action:  Go 0 of 3 selected

<input type="checkbox"/>	USER REQUIREMENTS
<input type="checkbox"/>	UserRequirements object
<input type="checkbox"/>	UserRequirements object
<input type="checkbox"/>	UserRequirements object

3 user requirementss

## **13. Source Code:**

### **Main.html:**

```
<!DOCTYPE html>
{%load staticfiles%}
<html>
<head>
<title>home page</title>
<style>
* {
padding:0;
margin:0;
box-sizing: border-box;
}
body{
background: linear-gradient(to top,rgba(0,0,0,0.5)50%,rgba(0,0,0,0.5)50%),url("{% static
'url/images/img2.jpg' %}");
background-position:center;
background-size:cover;
height: 109vh;
background-repeat:no-repeat;
font-family:sans-serif;
}
.menu-bar
{
text-align: center;
color:#7364ef;
}
.menu-bar ul
{
display: inline-flex;
list-style: none;
color: #fff;
}
.menu-bar ul li{
width: 100px;
margin: 10px;
padding: 6px;
}
.menu-bar ul li a
{
text-decoration: none;
color: #fff;
}
.sub-menu{
display:none;
}
```

```

.menu-bar ul li:hover .sub-menu
{
display: block;
position: relative;
margin-top: 15px;
}
.menu-bar ul li:hover .sub-menu ul
{
display: block;
margin: 10px;
}
.logo{
color: #ff7200;
font-size: 35px;
font-family: Arial;
padding-left: 20px;
float: left;
padding-top: 10px;
}
.scroll{
padding: 100px;
margin-top: 1500;
margin-bottom: 500;
font-size: 70px;
font-family: Arial, Helvetica, sans-serif;
font-style: italic;
color: #fff;
font-weight: 100;
}
#caption{
font-size: 50px;
font-family: Georgia, 'Times New Roman', Times, serif;
}
</style>
</head>
<body>
<div class="logo">
{% load static %}
</div>
<b>
<div class="menu-bar">
<ul>
<li class="active"><i class="fa fa-home"></i><a href="{% url 'home'
%}">Home</a></li>
<li><a href="{% url 'about' %}">About us</a></li>
<li><a href="#">Event</a>
<div class="sub-menu">

```

```

<ul>
<li><a href={% url 'wedding' %}>Wedding</a></li>
<li><a href={% url 'birthday' %}>Birthday</a></li>
<li><a href={% url 'babyshower' %}>Baby shower</a></li>
</ul>
</div>
</li>
<li><a href={% url 'contact' %}>Contact</a></li>
<!-- <li><a href={% url 'sign_up' %}>signup</a></li>-->
<li><a href={% url 'loginaction' %}>Logout</a></li>
<li><a href={% url 'feedbackview' %}>Feedback</a></li>
<li><a href={% url 'change_pass' %}>Changepassword</a></li>
</ul></div>
</b>
<div class="scroll">
<marquee scrollamount="15px" direction="right" width="200%">WELCOME TO RJ
EVENTS</marquee>
<marquee scrollamount="15px" id="caption">create your memories</marquee>
</div>
</body>
</html>

```

### views.py (main):

```

from django.shortcuts import render
from .models import UserRequirements,FeedBack
from django.shortcuts import render,HttpResponseRedirect
from django.contrib import messages
from django.core import serializers
from django.contrib.auth import update_session_auth_hash
from django.contrib.auth.forms import PasswordChangeForm
# Create your views here.
def main(request):
    return render(request,"main.html")
def birthday(request):
    return render(request,"birthday.html")
def babyshower(request):
    return render(request,"babyshower.html")
def register(request):
    if request.method == 'POST':
        name = request.POST['Fullname']
        mobile = request.POST['phone']
        email = request.POST['EMAIL']
        address = request.POST['Address']
        budget = request.POST['payment']
        requirements = request.POST['req']
        theme = request.POST['Theme']
        messages.success(request,'data registered successfully')

```

```

        obj= UserRequirements()
        obj.name= name
        obj.mobile = mobile
        obj.email = email
        obj.address = address
        obj.budget = budget
        obj.requirements = requirements
        obj.theme = theme
        obj.save()
    return render(request,"register.html")
def home(request):
    return render(request,"home.html")
def wedding(request):
    return render(request,"wedding.html")
def feedbackview(request):
    if request.method=='POST':
        Name=request.POST['name']
        Email=request.POST['email']
        feedback=request.POST['feedback']
        messages.success(request,'Feedback Submitted successfully')
        obj=FeedBack()
        obj.Name=Name
        obj.Email=Email
        obj.feedback=feedback
        obj.save()
        data = serializers.serialize("python",FeedBack.objects.all())
        my_dict={
            'data': data,
        }
    return render(request,'feedback.html',context=my_dict)
def change_pass(request):
    if request.user.is_authenticated:
        if request.method == 'POST':
            fm = PasswordChangeForm(user=request.user,data=request.POST)
            if fm.is_valid():
                fm.save()
                update_session_auth_hash(request,fm.user)
                messages.success(request,'Password changed successfully !!!!!')
            return HttpResponseRedirect('login')
        else:
            fm = PasswordChangeForm(user=request.user)
            return render(request,'changepass.html',{'form':fm})
    else:
        return HttpResponseRedirect('login')
def contact(request):
    return render(request,"contact.html")

```



## 14. Web Pages:

### 14.1 Sign Up Page:

Activities Visual Studio Code Wed 8:47 AM Sign Up form — Mozilla Firefox

Sign Up form

127.0.0.1:8000/signup/ 110%

Most Visited Getting Started Week 1 Interview Que... 10 Sorting Algorithms... Apply Confirmation Box Model HackerRank Advance... Other Bookmarks

## Sign Up Form

Username:

First name:

Last name:

Email:

Password:

Password confirmation:

[login](#)

### 14.2 Login Page:

Activities Visual Studio Code Tue 5:59 AM Mozilla Firefox


127.0.0.1:8000/

127.0.0.1:8000

Most Visited Getting Started Week 1 Interview Que... 10 Sorting Algorithms... Apply Confirmation Box Model HackerRank Advance... Other Bookmarks

To visit this website please login here

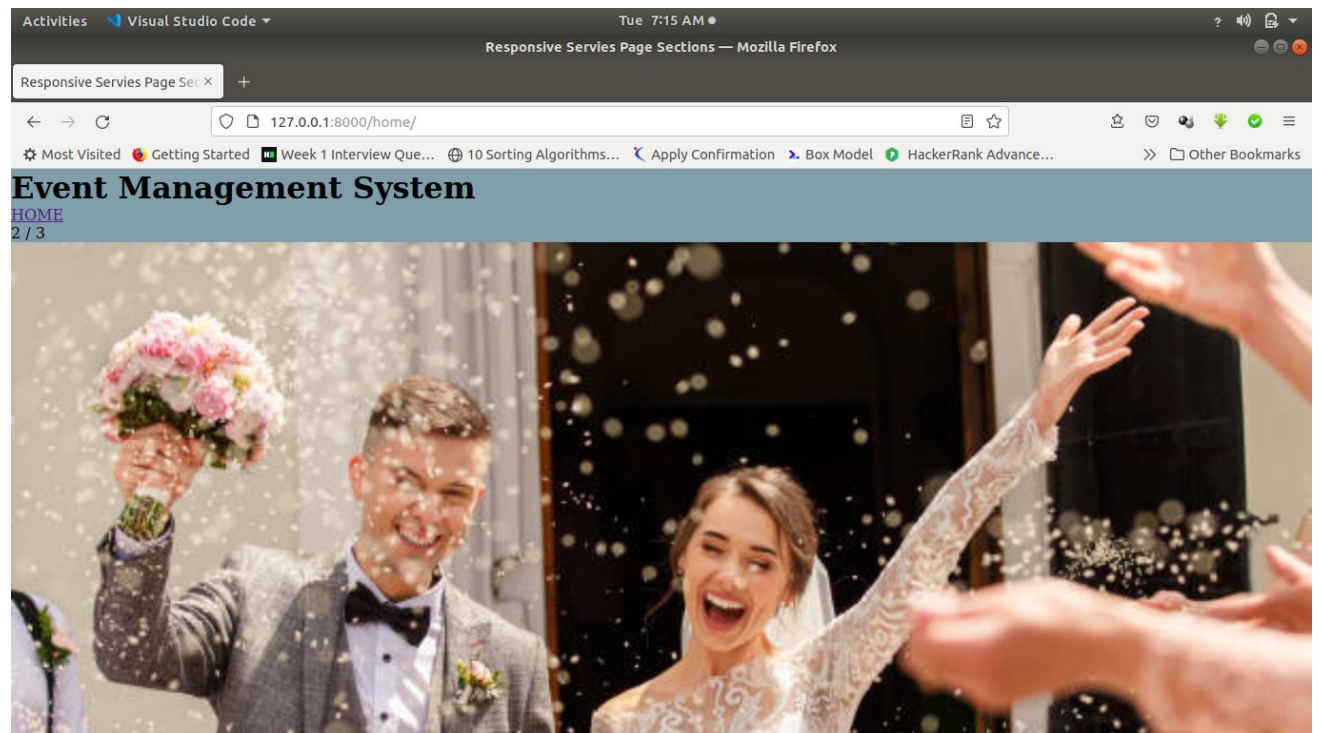
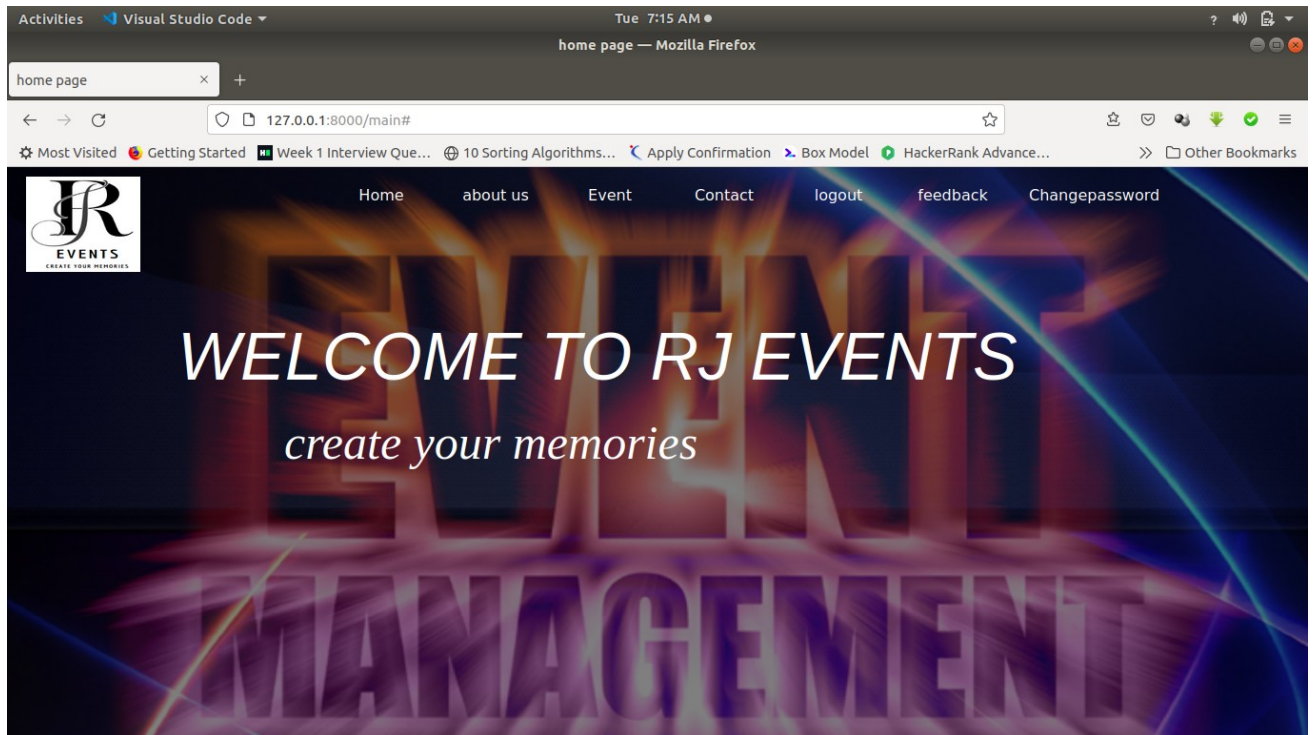
## Login Form



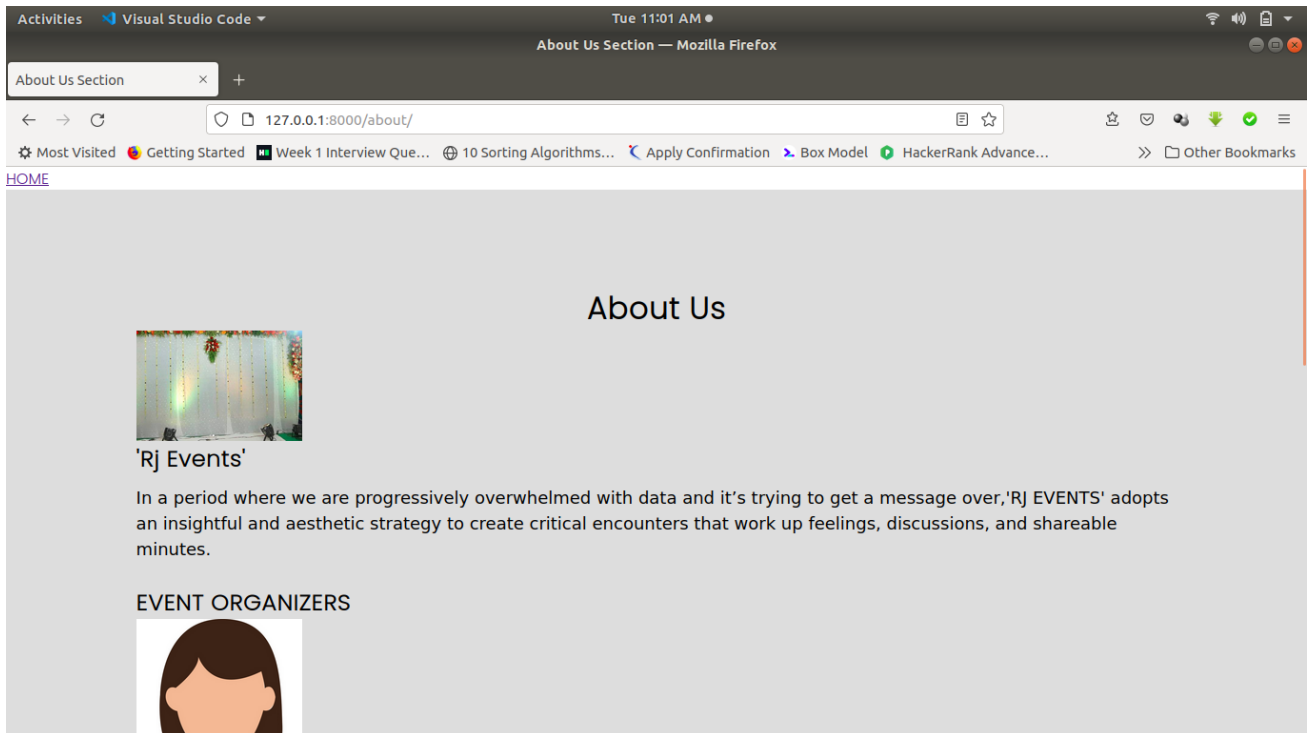
Username:

Password:

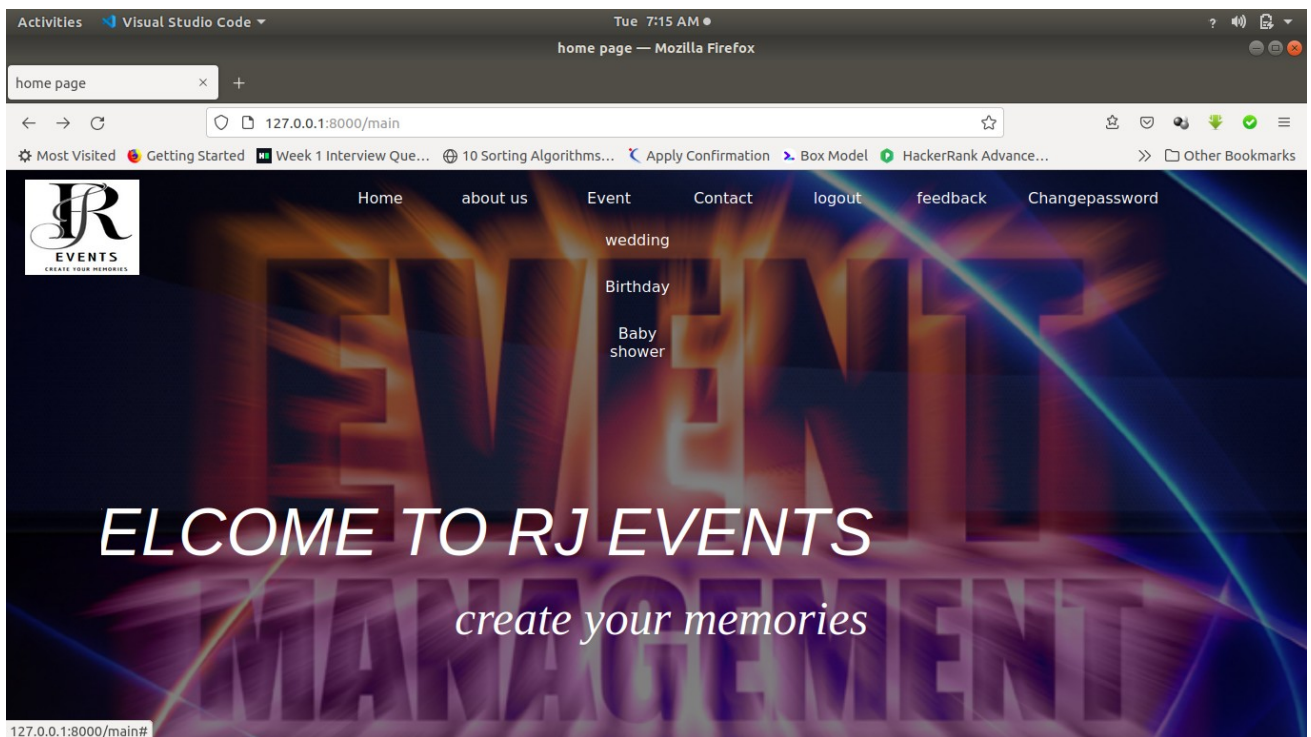
### 14.3 Home Page:



## 14.4 About Us Page:

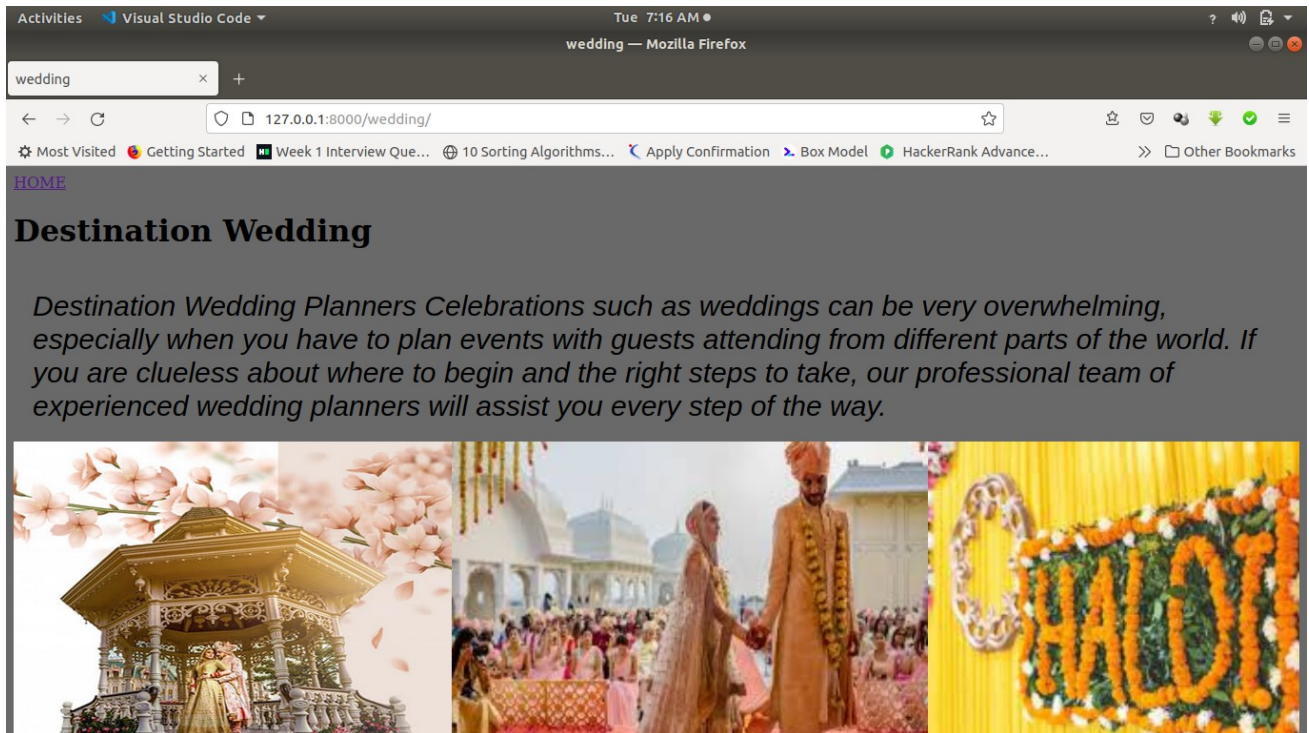


## 14.5 Events Page:

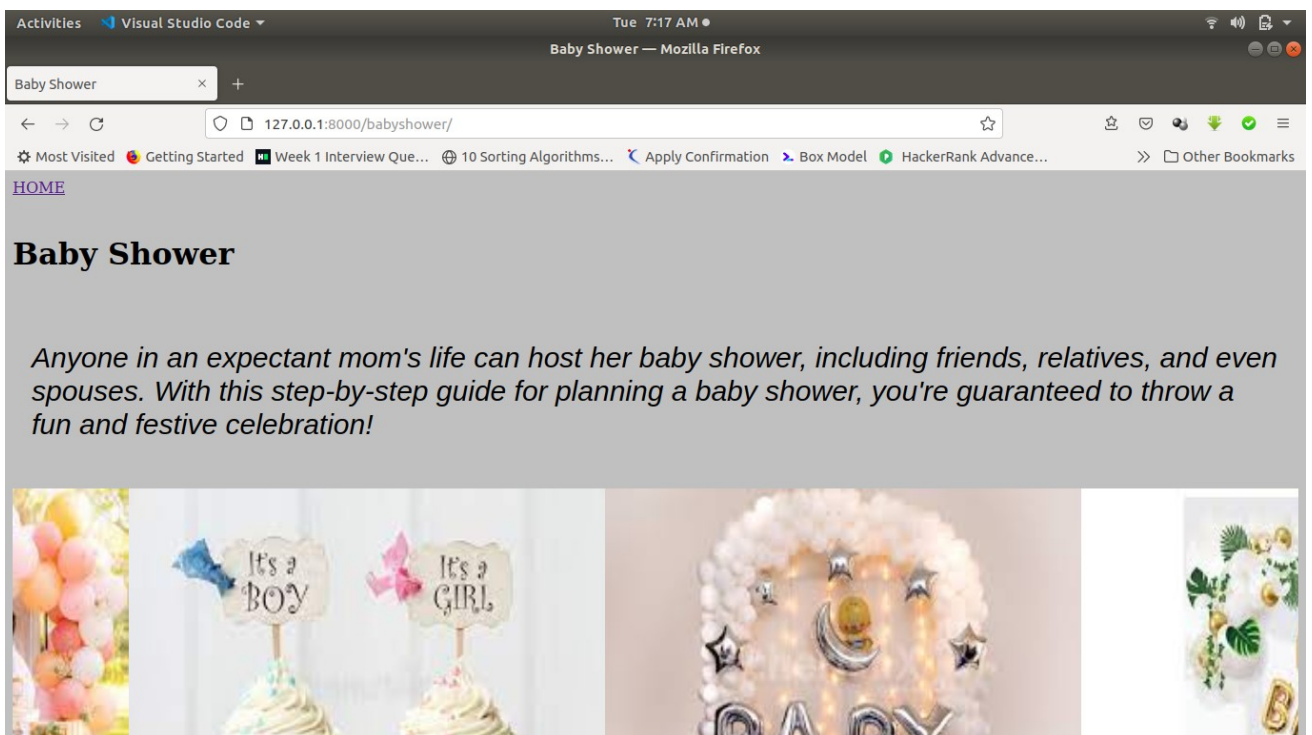




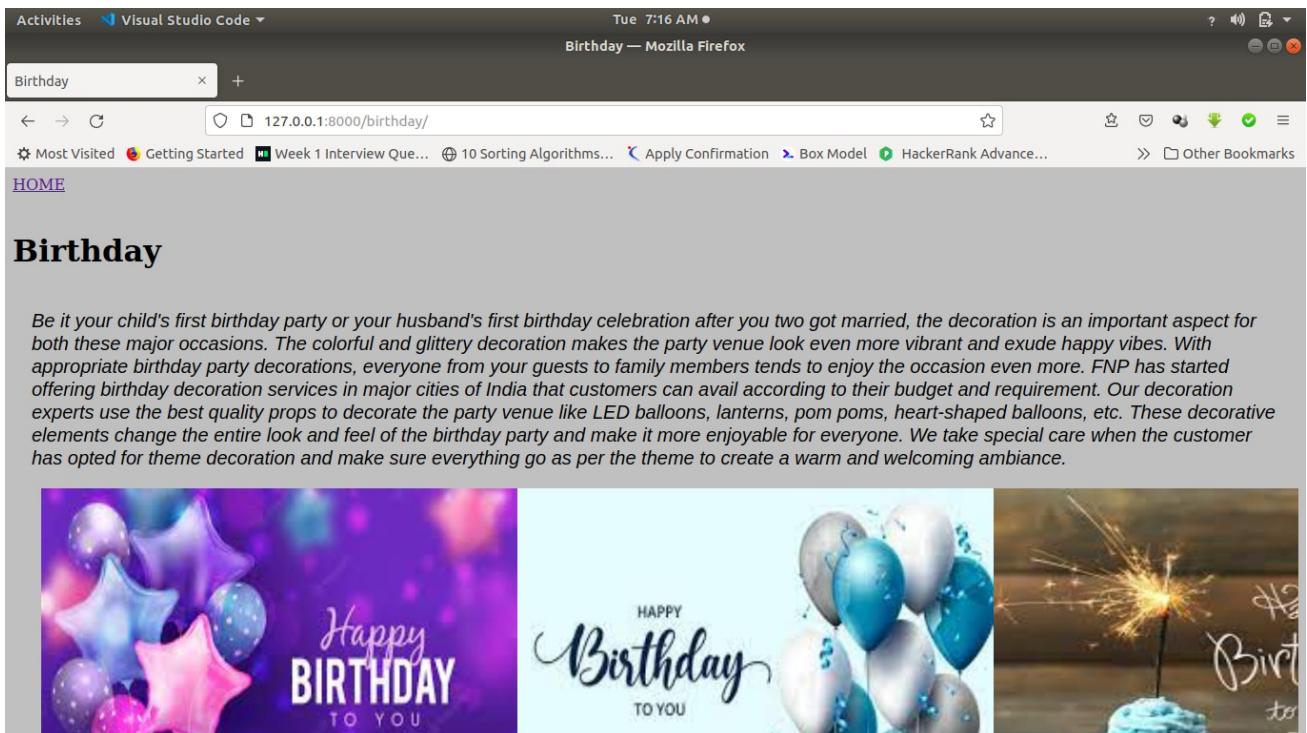
## 14.6 Wedding Page



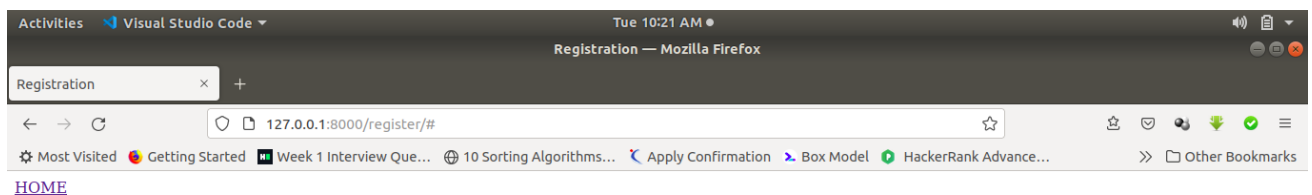
## 14.7 Baby Shower Page



## 14.8 Birthday Page



## 14.9 Registration Page



### Registration Form

First Name	:	<input type="text" value="Enter FullName"/>
Mobile No	:	<input type="text" value="Ex: 9502836885"/>
Email Id	:	<input type="text" value="jack05@gmail.com"/>
Address	:	<input type="text" value="Event Address"/>
Budget	:	<input type="text" value="choose"/>

## 14.10 Contact Us Page

Activities Visual Studio Code Wed 8:57 AM

Responsive Contact Us Page — Mozilla Firefox

Responsive Contact Us Page × +

127.0.0.1:8000/contact/

Most Visited Getting Started Week 1 Interview Que... 10 Sorting Algorithms... Apply Confirmation Box Model HackerRank Advance... Other Bookmarks

### Contact Us

Address

RGUKT RK Valley,  
Idupulapaya,  
Vempalli  
Kadapa  
516330

Phone

9494308680  
9502836884

Email

r170427@rguktrkv.ac.in  
r170415@rguktrkv.ac.in

Send Message

Full Name

Ex: Jahnvi

Email

Ex: ekotijahnavi0504@gmail.com

Type Your Message...

submit

## 14.11 Feedback Page

Activities Visual Studio Code Wed 8:48 AM

feedback form — Mozilla Firefox

feedback form × +

127.0.0.1:8000/feedback/

Most Visited Getting Started Week 1 Interview Que... 10 Sorting Algorithms... Apply Confirmation Box Model HackerRank Advance... Other Bookmarks

### Feedback Form

Name: Ex: Jahnvi

Email: Ex: ekotijahnavi0504@gmail.com

Feedback:

Submit Feedback

Reset

fetch and refresh

Name	Email	feedback
mouni	chivakalamounika@gmail.com	RJ Events managing and organizing is fantastic.We created a lot beautiful memories.Friends You can also approach RJ Events.
geetha	geethu@gmail.com	very Nice
geetha	geethu@gmail.com	very Nice
Jayasree	ekotijayasree@gmail.com	fantastic

## 14.12 Change Password Page

Activities Visual Studio Code Tue 7:19 AM • change password — Mozilla Firefox

change password × +

← → ↻ 127.0.0.1:8000/changepass/ ☆

⚙ Most Visited 🌈 Getting Started 📺 Week 1 Interview Que... 🌐 10 Sorting Algorithms... 🔗 Apply Confirmation 📦 Box Model 🟢 HackerRank Advance... >> 📁 Other Bookmarks

### Change Password

Old password:

New password:

New password confirmation:

## 15. References:

1. [https://www.udemy.comshare101soi3@8EC6KVhGb09rQmorwqy6lD6GET0ufkpd\\_V64dUG4wYhbyCgileSXLuiZoUz1BbBazw==/](https://www.udemy.comshare101soi3@8EC6KVhGb09rQmorwqy6lD6GET0ufkpd_V64dUG4wYhbyCgileSXLuiZoUz1BbBazw==/)
2. [https://www.w3schools.com/django/django\\_intro.php](https://www.w3schools.com/django/django_intro.php)
3. <https://www.digitalocean.com/community/tutorials/how-to-install-the-django-web-framework-on-ubuntu-18-04>
4. <https://stackoverflow.com/questions/36797051/django-login-error-attempt-to-write-a-readonly-database>