

**“GOLDEN MEMORIES”**

**BACHELOR OF TECHNOLOGY**  
in  
**COMPUTER SCIENCE AND ENGINEERING**



**RGUKT**  
**Rajiv Gandhi University of Knowledge Technologies**  
**R.K.VALLEY**

**Submitted by**  
**C.Meghana - R170035**

**Under the Esteemed guidance of**  
**Mrs.Rajeswari**  
**Guest Faculty**  
**RGUKT RK Valley.**

# **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 “GOLDEN MEMORIES” submitted by C.Meghana(R170035) under our guidance and supervision for the partial fulfilment for the degree Bachelor of Technology in Computer Science and Engineering during the academic semester 2 2022-2023 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**

Mrs. Rajeswari  
Guest Faculty  
RGUKT, RK Valley

### **Head of the Department**

Mr.Satyanandaram.N  
HOD Of CSE  
RGUKT, RK Valley

## **ACKNOWLEDGEMENT**

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts success. We are extremely grateful to our respected Director, Prof. K. SANDHYA RANI for fostering an excellent academic climate in our institution.

We also express our sincere gratitude to our respected Head of the Department Mr. N SATYANANDARAM for his encouragement, overall guidance in viewing this project as a good asset and effort in bringing out this project. We would like to convey thanks to our guide at college Mrs. RAJESWARI for her guidance, encouragement, co-operation and kindness during the entire duration of the course and academics. Our sincere thanks to all the members who helped us directly and indirectly in the completion of project work. We express our profound gratitude to all our friends and family members for their encouragement.

# **TABLE OF CONTENTS**

1. ABSTRACT-----	5
2. INTRODUCTION-----	6
3.OBJECTIVES-----	7
4. SCOPE-----	7
5. ADVANTAGES-----	7
6. DISADVANTAGES-----	7
7. MODULES-----	8
8. TECHNOLOGIES-----	8-9
9. CONTEXT DIAGRAM-----	10
10.SYSTEM WIDE REQUIREMENTS-----	10-13
11.FUNCTIONAL REQUIREMENTS-----	13-14
12.STEPS-----	15
13.ADMIN-----	16-17
14.SOURCE CODE-----	18-21
15.WEB PAGES-----	22-27
16.REFERENCES-----	27

## **ABSTRACT**

Now-a-days people are always interested and excited about making memories and binding them in the form of albums. Every small event can become a very good memory if we store them. The small happiness that people want to save forever can be hold in the form of a very beautiful hard disk that contain everything about them.

Every event hold a better place in everyone's heart. Some cannot have such memories because they can't able to do them because of financial problems or not knowing about such event managers. Without going around people can search online and book the events. This is a project which is budget friendly and pre-booking are available for all types of events.

## **INTRODUCTION**

An event can be described as a public assembly for the purpose of celebration, education, marketing or reunion. Events can be classified on the basis of their size , type and context. An event can be social event like a birthday party , engagement , Wedding etc.. or an education and career event like an education fair, job fair , workshop, contest etc..

Event Management is the process of analysing, planning, marketing, producing and evaluating an event. It is a different way of promoting a product , service or idea. If an event is managed efficiently and effectively. It can be used as a very powerful promotional tool to launch or market a product or service. Event management requires certain core values to be deployed to every element, process and decision to justify professional approach and achieve effective and efficient results.

## **OBJECTIVE**

Objectives are essentially aims broken down into specific targets, to facilitate event delivery and evaluation. Events can deliver beneficial impacts and outcomes both for the organisers and the host community, in addition to other stakeholders such as participants, spectators, sponsors and the media.

## **SCOPE**

Through this forum you can discuss and share event management and planning ideas with other event professionals. You can promote your business and events, upload your events photos and event videos. You can also create groups, blogs and can even do live chat with other event planning professionals.

## **ADVANTAGES**

- Easier event management
- Saves time
- Increase engagement
- Cut costs
- Analyse and improve
- Automate your event
- Display a greater level of professionalism.
- Event automation.

## **DISADVANTAGES**

- Long hours
- Travel Requirements
- Customer Demands
- High Stress
- Low starting salaries.

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

- ▯ Event Module: Managing all the event
- ▯ Sign Up Module: Managing the signup credentials.
- ▯ Login Module: Managing the login details.
- ▯ Home Module: Managing the Web Page Details.
- ▯ About Us Module: Managing the managers details.
- ▯ Registration Module: Managing the registration details.

## **Technologies:**

- ▯ HTML
- ▯ CSS
- ▯ Python
- ▯ JavaScript
- ▯ Django
- ▯ DB SQLite

### **HTML:**

The HTML or HyperText Markup Language is the standard markup language for documents designed to be displayed in a web browser. It is often assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript. It is also used to create electronic documents that are generally displayed on the World Wide Web.

### **CSS:**

Cascading Style Sheets or CSS is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.



## **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.

## **JavaScript:**

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

## **Django:**

Django is a high-level Python web framework that 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.

## **DB SQLite:**

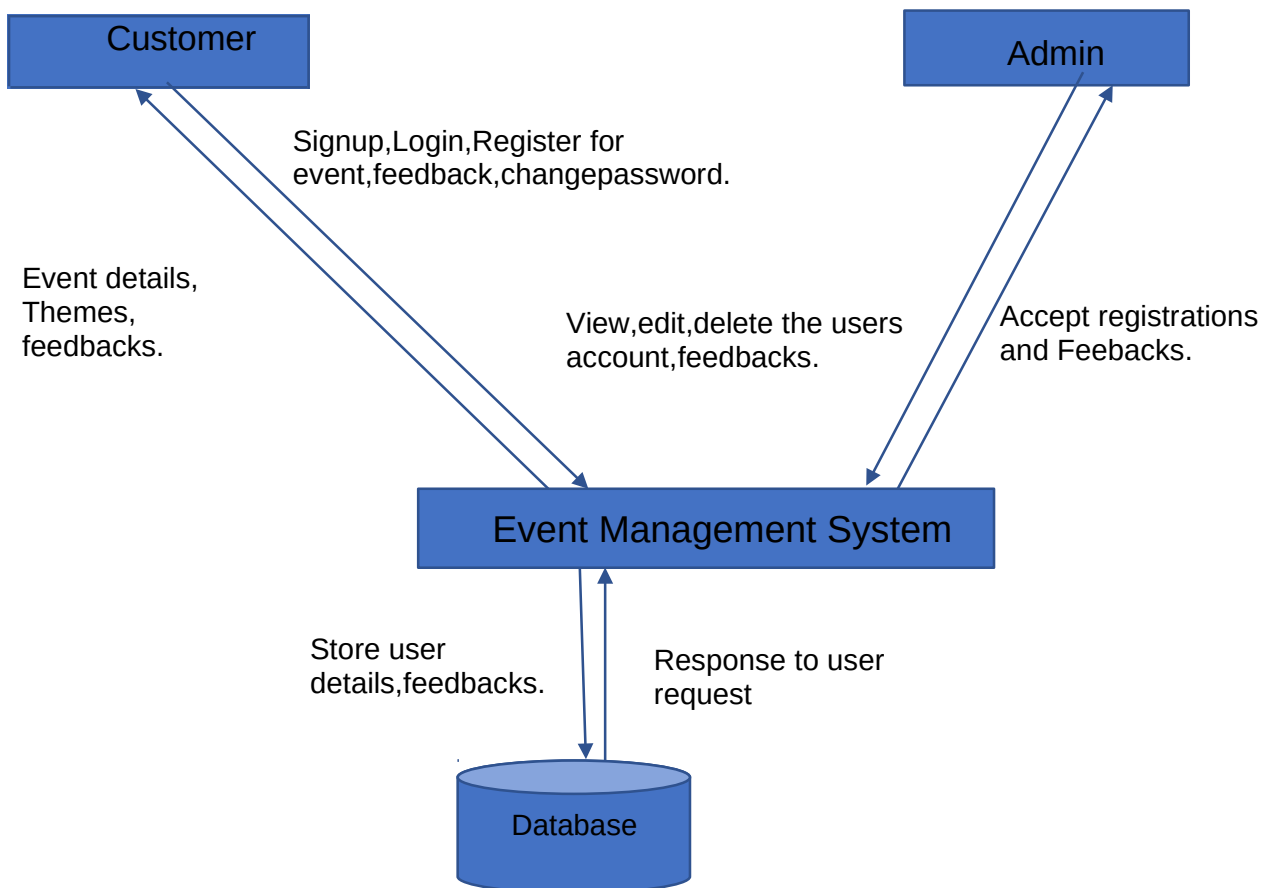
SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine.

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.

## **System in Context:**

This GOLDEN MEMORIES 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:**

“GOLDEN MEMORIES” is a multi-user system and it provides activities

associated with day-to-day questions.

### **1.The most crucial events are:**

1. To use the application the user have to register himself with a personal email , name and the password for the future uses.
2. Customer have to login using the users personal username and password.
3. Among the different events available customer have to choose the event needed.
4. After selecting event, registration of specified event is required.
5. The details about budget and event need to be specified in the specified event.
6. After this whole process the customer have to submit registration form.

**The below table provides a set of user visible events that define the functionalities that are in 'GOLDEN MEMORIES'.**

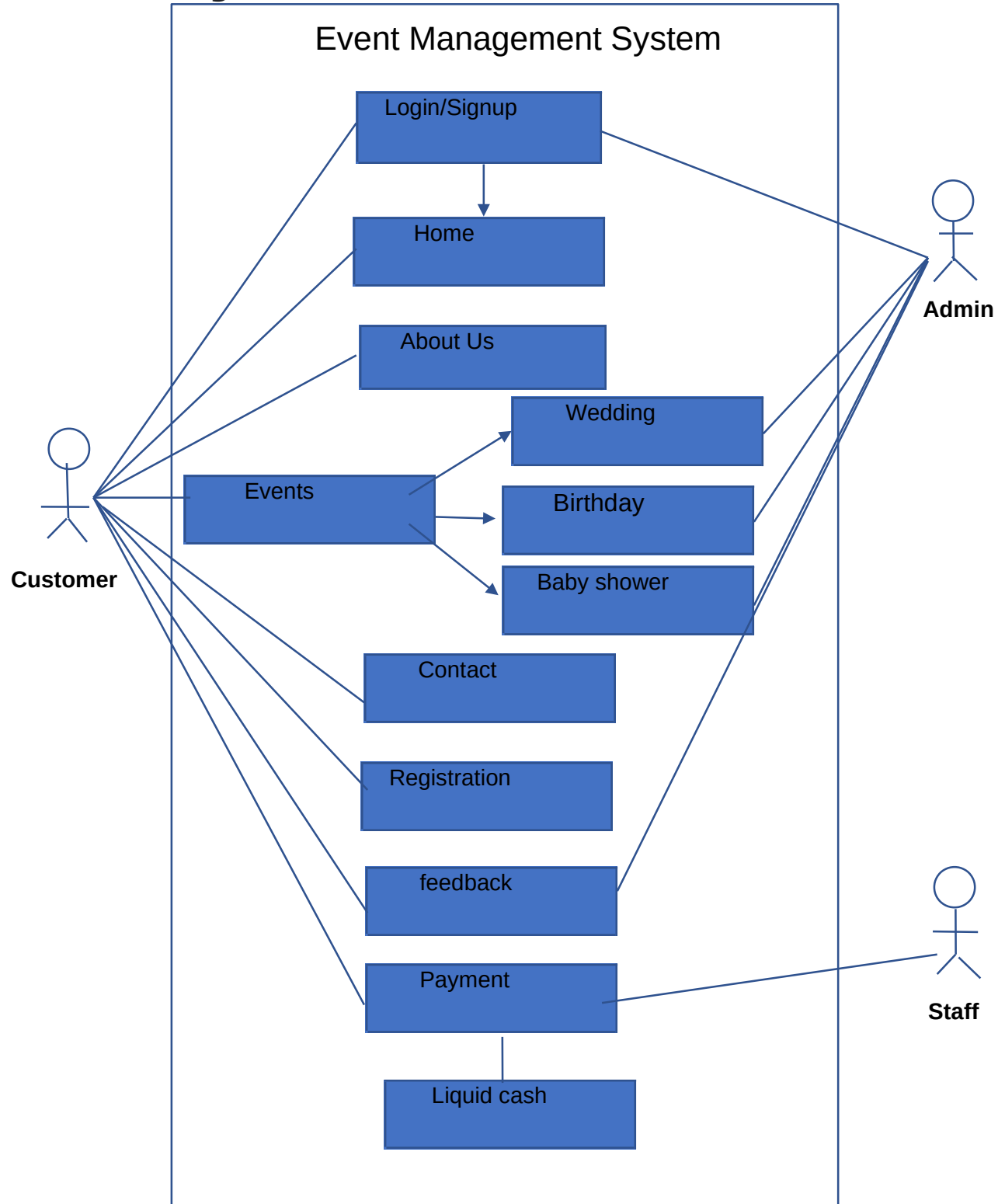
	Actor	Action	Object	Frequency	Arrival Pattern	Response
1.	Customer	onclick	Signup	1/day	Episodic	It asks details about username first name last name email along with the password of the user.
2.	Customer	onclick	Login	1/day	Episodic	It asks username password
3.	Customer	onclick	Main	1/day	Episodic	It displays all modules in the software that

						tells about different events.
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.
--	--	--	--	--	--	---------------------------

## **Functional Requirements:**

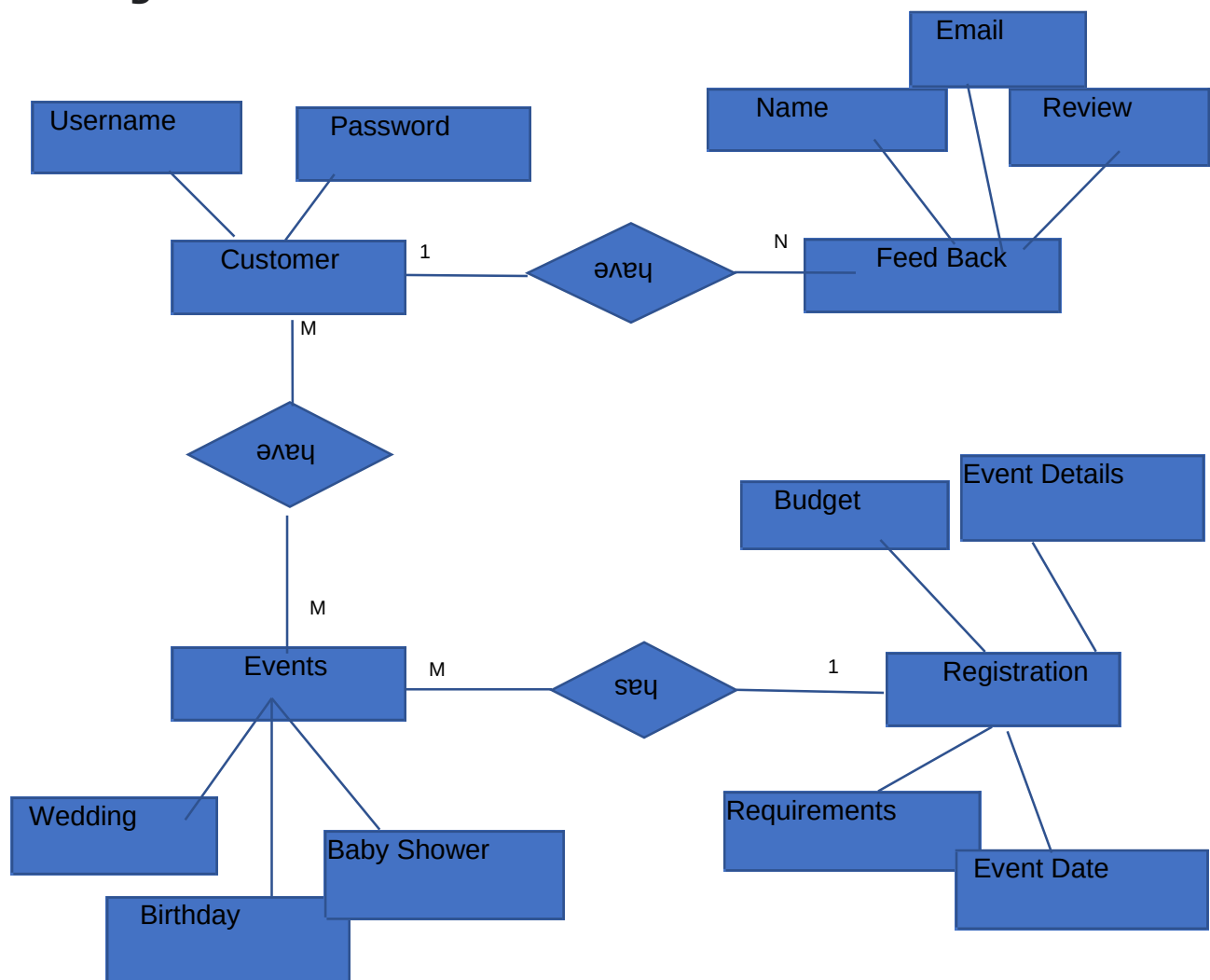
### **Use Case Diagram:**



## Use Case Overview:

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

## ER Diagram:



## 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
```

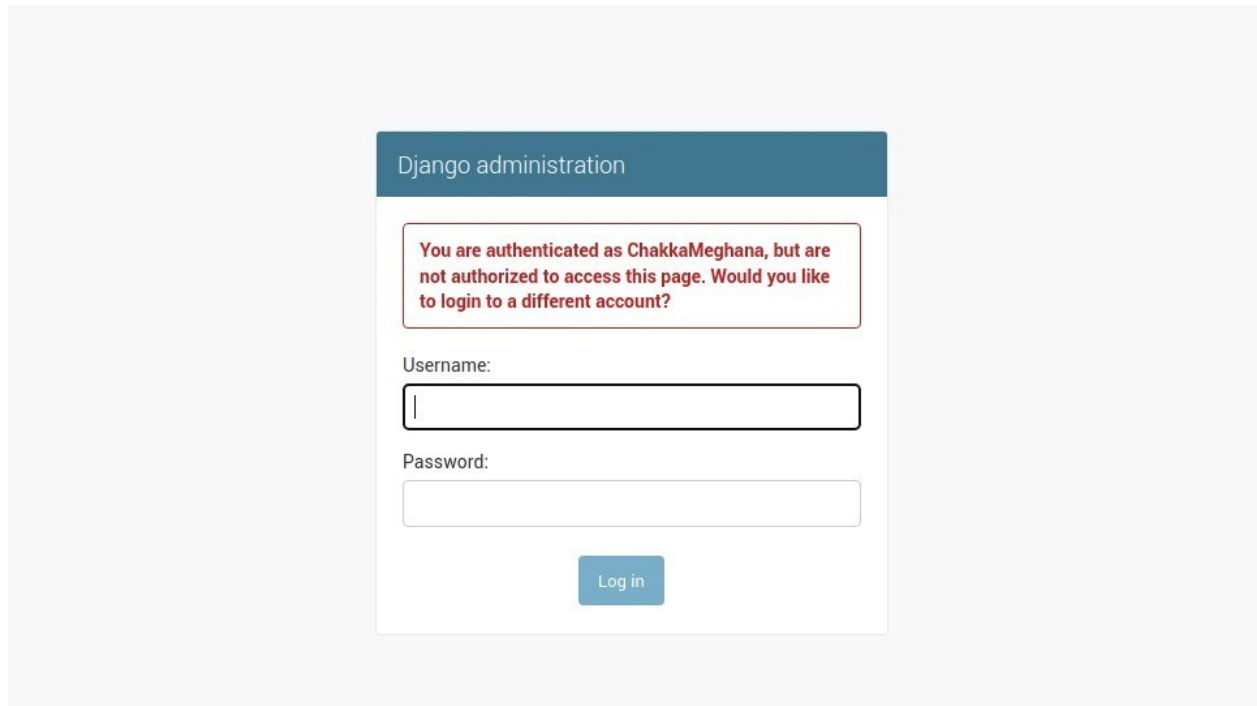
Make sure that run this command in the particular directory where this manage.py is present. This command will generate an url <http://127.0.0.1:8000/> use ctrl+click on the url to open in the browser.

8.Send the request.

## Admin Page:

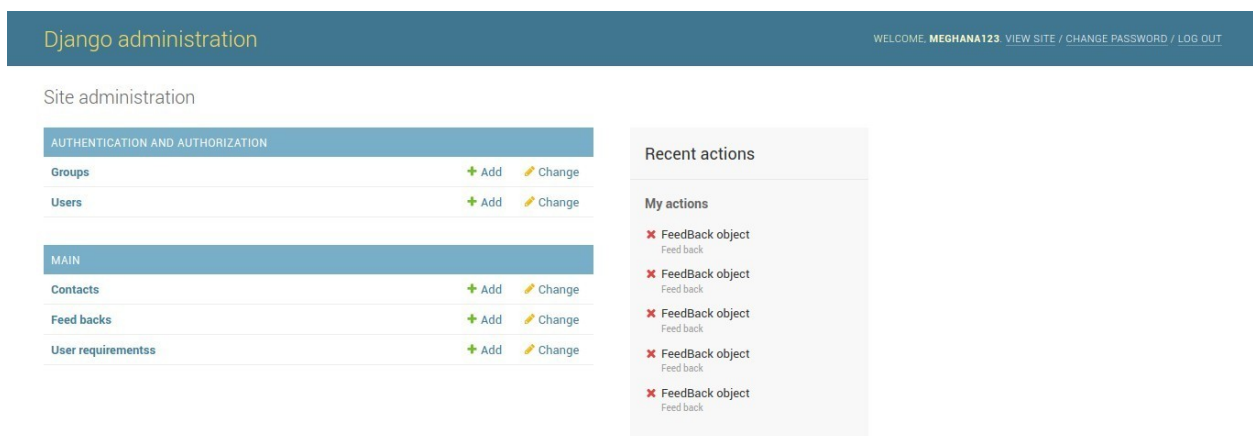
Django provides a default admin interface which can be used to perform create, read, update and delete operations on the model directly. It reads set of data that explain and gives information about data from the model, to provide an instant interface where the user can adjust contents of the application.

## Admin Login Page:



The image shows the Django administration login page. At the top, there is a blue header with the text "Django administration". Below this, a red-bordered box contains a warning message: "You are authenticated as ChakkaMeghana, but are not authorized to access this page. Would you like to login to a different account?". Underneath the warning, there are two input fields: "Username:" and "Password:". A blue "Log in" button is positioned below the password field.

## Super User Creation:



The image displays the Django administration site interface. At the top, a blue header bar contains the text "Django administration" on the left and "WELCOME, MEGHANA123 VIEW SITE / CHANGE PASSWORD / LOG OUT" on the right. Below the header, the page is titled "Site administration". The main content area is divided into two sections: "AUTHENTICATION AND AUTHORIZATION" and "MAIN".

AUTHENTICATION AND AUTHORIZATION	
Groups	<a href="#">+ Add</a> <a href="#">Change</a>
Users	<a href="#">+ Add</a> <a href="#">Change</a>

MAIN	
Contacts	<a href="#">+ Add</a> <a href="#">Change</a>
Feed backs	<a href="#">+ Add</a> <a href="#">Change</a>
User requirements	<a href="#">+ Add</a> <a href="#">Change</a>

On the right side of the page, there is a "Recent actions" section. It contains a list of actions under the heading "My actions". Each action is preceded by a red "X" icon and followed by the text "Feed back".



Select feed back to change

[ADD FEED BACK](#) +Action:   0 of 1 selected☐ FEED BACK☐ FeedBack object

1 feed back

Logged out

Thanks for spending some quality time with the Web site today.

[Log in again](#)

## **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
GOLDEN MEMORIES</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')

```

```

23
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()

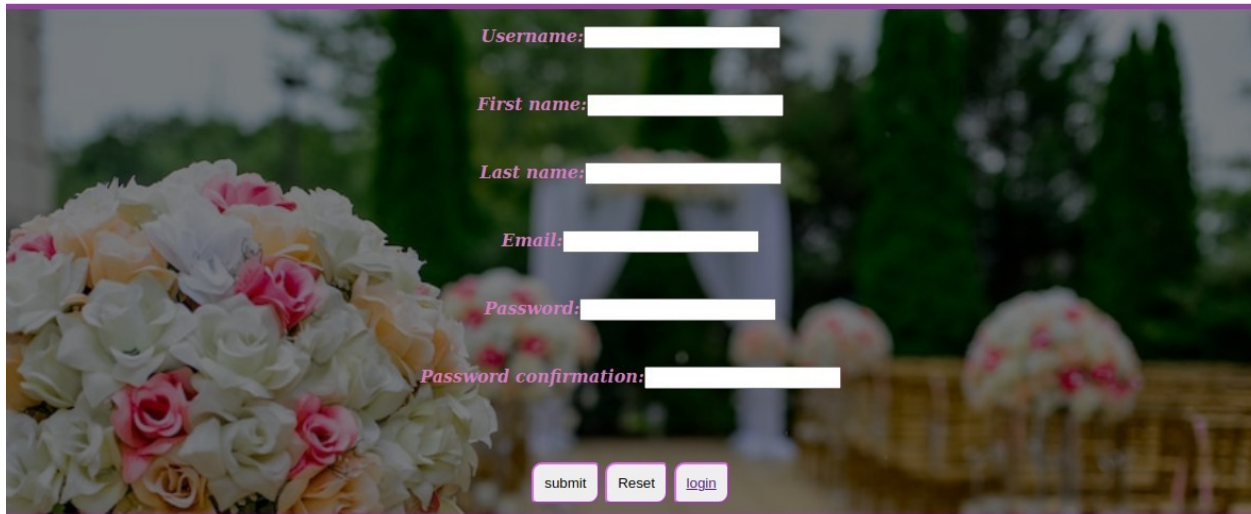
return render(request,"register.html")
def home(request):
    return render(request,"home.html")
def wedding(request):
    return render(request,"wedding.html")
def feedbackview(request):
    if request.method=='POST':
        Name=request.POST['name']
        Email=request.POST['email']
        feedback=request.POST['feedback']
        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")

```

## Web Pages:

### Sign Up Page:

#### Sign Up Form

A sign-up form with a background image of a wedding bouquet. The form includes fields for Username, First name, Last name, Email, Password, and Password confirmation. At the bottom are buttons for submit, Reset, and login.

Username:

First name:

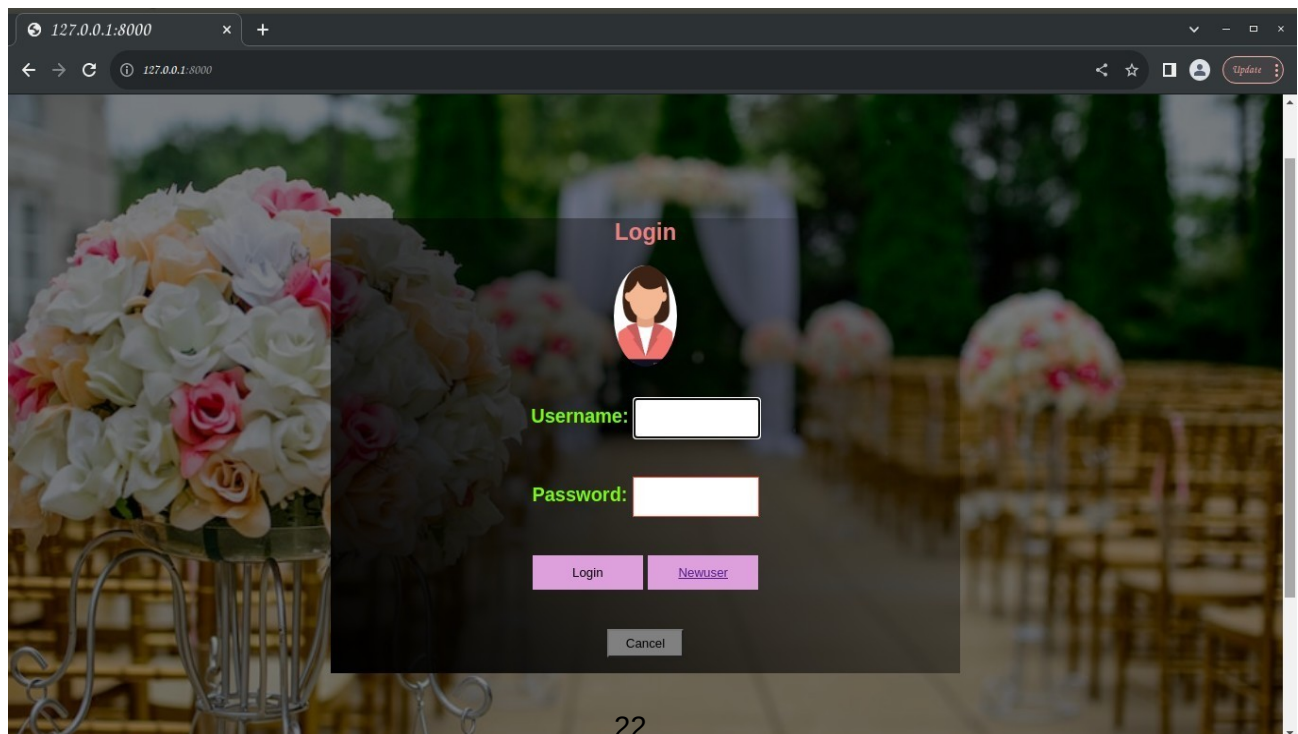
Last name:

Email:

Password:

Password confirmation:

### Login Page:

A login page shown within a browser window. The browser address bar shows '127.0.0.1:8000'. The login form has a title 'Login' and a user icon. It includes fields for Username and Password, and buttons for Login, Newuser, and Cancel.

127.0.0.1:8000

127.0.0.1:8000

Update

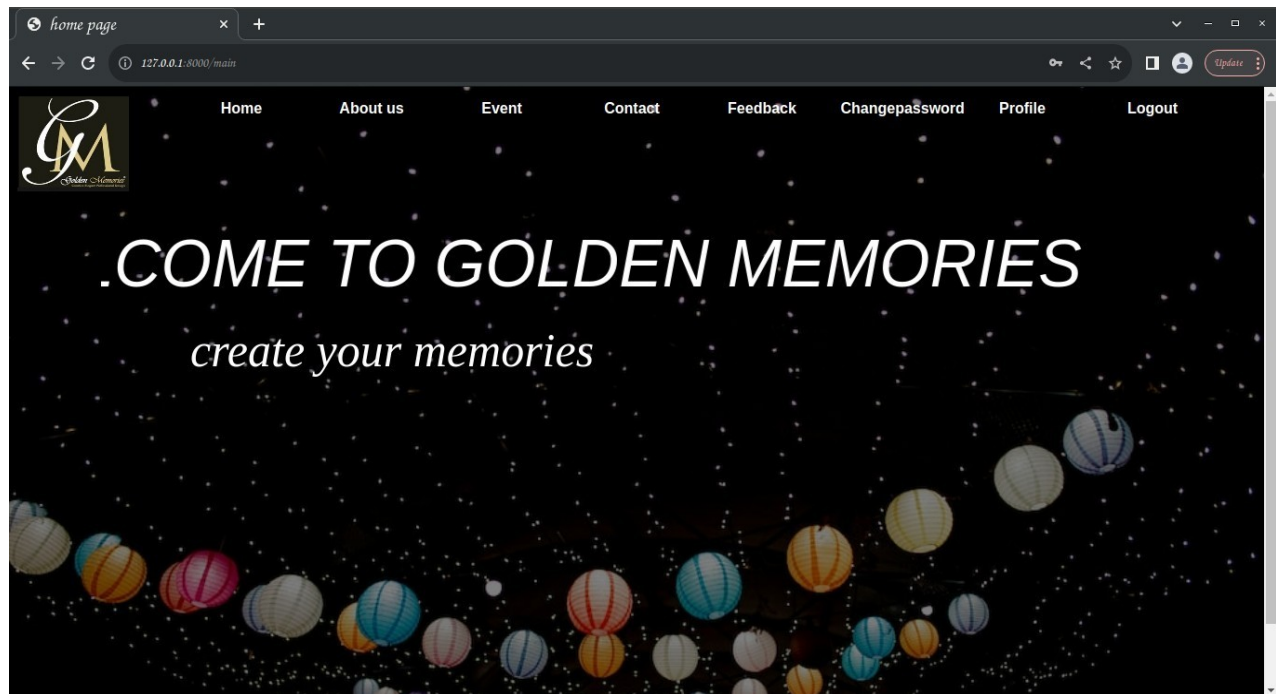
Login

Username:

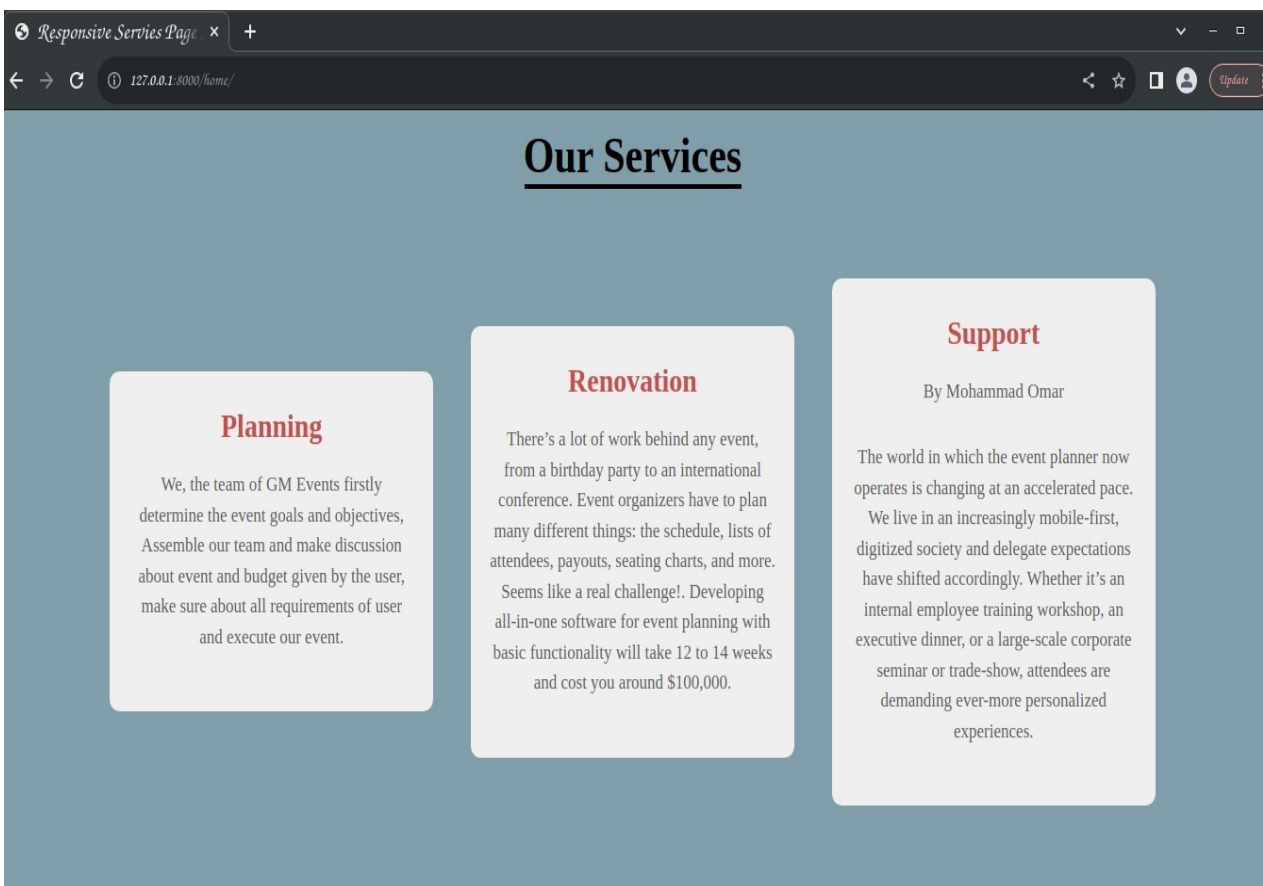
Password:

22

## Home Page:

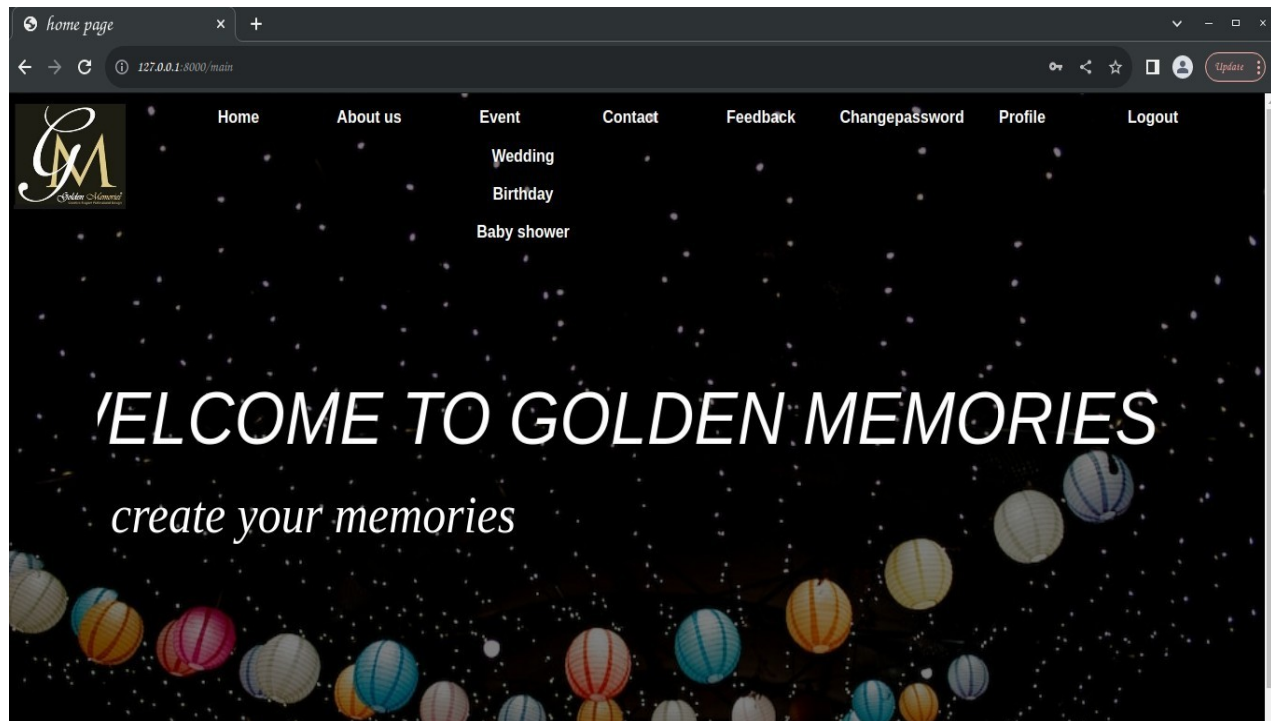


## Services Page:

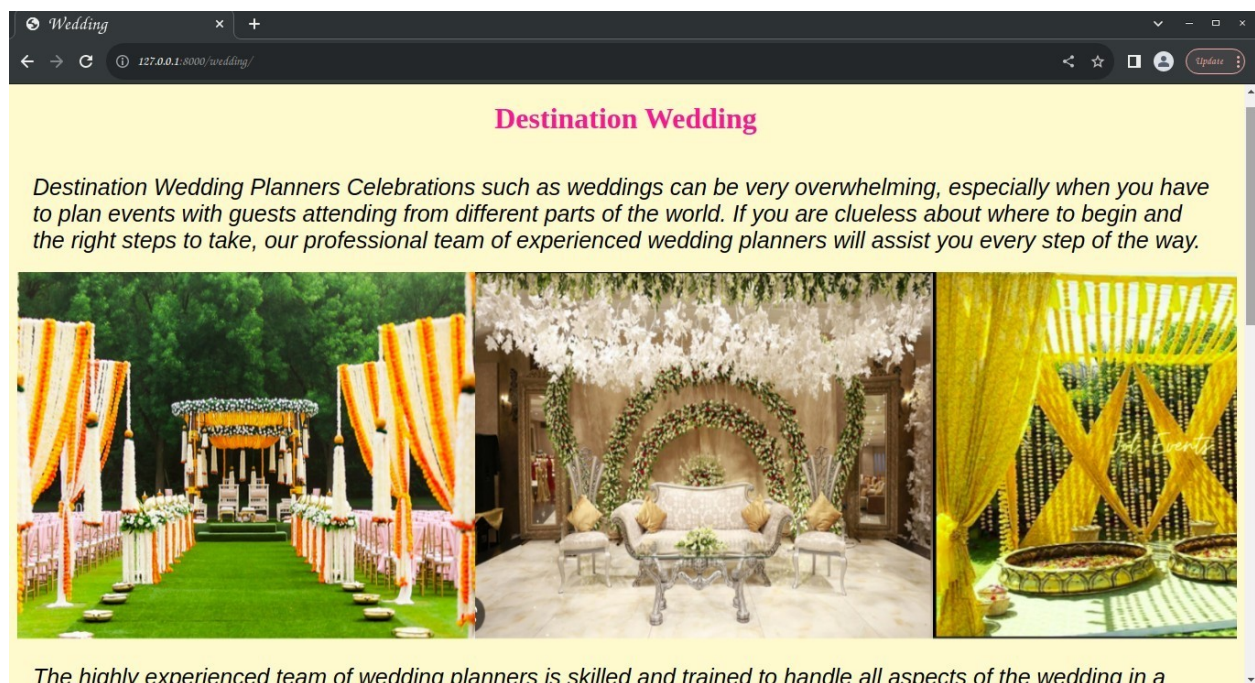




## Events Page:



## Wedding Page:





## **Birthday Page:**


Birthday

127.0.0.1:8000/birthday/

Update

### Birthday

*Be it your child's first birthday party or your husband's first birthday celebration after you two got married, the decoration is an important aspect for both these major occasions. The colorful and glittery decoration makes the party venue look even more vibrant and exude happy vibes. With appropriate birthday party decorations, everyone from your guests to family members tends to enjoy the occasion even more. FNP has started offering birthday decoration services in major cities of India that customers can avail according to their budget and requirement. Our decoration experts use the best quality props to decorate the party venue like LED balloons, lanterns, pom poms, heart-shaped balloons, etc. These decorative elements change the entire look and feel of the birthday party and make it more enjoyable for everyone. We take special care when the customer has opted for theme decoration and make sure everything go as per the theme to create a warm and welcoming ambiance.*



## **Baby Shower Page:**


Baby Shower

127.0.0.1:8000/babyshower/

Update

### Baby Shower

*Anyone in an expectant mom's life can host her baby shower, including friends, relatives, and even spouses. With this step-by-step guide for planning a baby shower, you're guaranteed to throw a fun and festive celebration!*



## Contact Us page :

Responsive Contact Us | x +

127.0.0.1:8000/contact/

HOME

### Contact Us

**Address**  
RGUKT RK Valley,  
Idupulapaya,  
Vempalli  
Kadapa  
516330

**Phone**  
9676370538  
7337372714

**Email**  
r170035@rguktrkv.ac.in  
r170025@rguktrkv.ac.in

#### Send Message

Full Name

Email

Type Your Message...

## Feedback Page:

feedback form x +

127.0.0.1:8000/feedback/

HOME

### Feedback Form

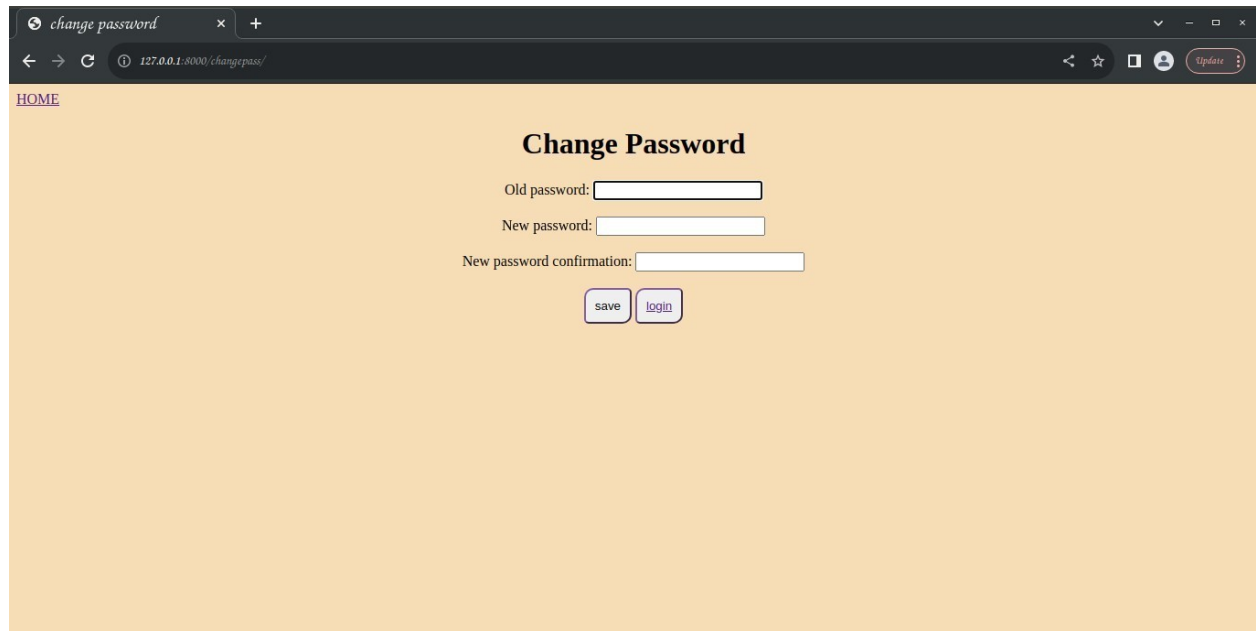
Name:

Email:

Feedback:

Name	Email	feedback
magi	janu@gmail.com	nice

## **Change Password Page:**



change password x +

← → ↻ ⓘ 127.0.0.1:8000/changepass/ ☆ □ ⓘ Update ⋮

[HOME](#)

### Change Password

Old password:

New password:

New password confirmation:

## **References:**

1. <https://stackoverflow.com/questions/36797051/django-login-error-attempt-to-write-a-readonly-database>
2. <https://www.digitalocean.com/community/tutorials/how-to-install-the-django-web-framework-on-ubuntu-18-04>.

-