

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

SURANA COLLEGE

#16, SOUTH END BENGALURU



DEPT. OF COMPUTER SCIENCE

ELEGANT EVENTS

By

Padmini Kustagi (17KXSB7063)

Under the guidance of
Asst. Prof. Mithili Devi

ELEGANT EVENTS project report submitted in partial
fulfilment of the requirements of VI semester BCA,
SURANA COLLEGE

OCT 2020

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

SURANA COLLEGE

#16, SOUTH END BENGALURU

Page | 2



DEPT. OF COMPUER SCIENCE

CERTIFICATE

This is to certify that the report titled ELEGANT EVENTS is a bona fide record of work done by PADMINI KUSTAGI (17KXSB7063) of Surana College Bangalore, in partial fulfilment of the requirements of VI Semester BCA during the year 2020.

Head of the Department

Project Guide

Valued by:

Name	: Padmini Kustagi
Register Number	: 17KXSB7063
Examination center	: SURANA COLLEGE (SOUTH END)
Date of Exam	: 10/10/2020

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

INDEX

Page | 3

CONTENTS SL No.	Topic	Page no.
1	ACKNOWLEDGEMENT	4
2	PROJECT SYNOPSIS	5
3	PROJECT MODULES	6
4	SYSTEM ENVIRONMENT	7
5	DATA FLOW DIAGRAM	8
6	ER DIAGRAM	9
7	DATABASE TABLES	10
8	SCREEN SHOTS	13
9	SOURCE CODE	17
10	TESTING	74
12	FUTURE ENHANCEMENT	75
13	CONCLUSION	75
14	BIBIOGRAPHY	75

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

ACKNOWLEDGMENT

Page | 4

The successful and final outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.

*I respect and thank our Head of Department (H.O.D) **Mr. SRINIVAS A** for providing us an opportunity to do the project work and giving us all support and guidance, which made us complete the project duly.*

*I owe my deep gratitude to our project guide **Mrs. MITHILI DEVI** who took keen interest on our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good application.*

*We are fortunate enough to get constant encouragement, support and guidance from all Teaching staff of Computer Science Department which helped us in successfully completing of our project work. I would like to thank **Mr. HANUMESH PUROHIT** for his constant support and helping hand with the project. I would also like to thank our friends and family who've been constantly supporting us with the work.*

Thank You

PADMINI KUSTAGI

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

SYNOPSIS

Page | 5

Project Title: Elegant Events

Objective:

This project focuses on creating an interactive website for an event management company and aims to provide services for social events, both formal and informal.

It aims to create an interface for event planning on the user end and provides sorted data on the backend.

Project Description:

The project is a web application containing tabs that offer Event management services, program to load relative content, careers at the company and Login authentication.

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

PROJECT MODULES

HOME PAGE:

The Home page consists of navigation to the tabs: Events, Services, Gallery, About Us, Contact Us, Join our team, Sign-in and Sign-up

Page | 6

EVENTS:

The description of Events and Ceremonies organized and managed by the company is displayed, along with redirection to booking and payment options.

SERVICES:

The description of Services offered and managed by the company is displayed, along with redirection to booking and payment options.

GALLERY:

Displays a variety of pictures from the events the company has organized and managed.

ABOUT US:

A description all about the company, its objectives and goals.

CONTACT US:

A form to accept order and event related queries from customers.

JOIN OUR TEAM:

A form to accept resumes from candidates who would like to join the company.

SIGN-IN:

A form to enter the website as a registered user or for admin login.

SIGN-UP:

A form to register as a user to the website.

DASHBOARD:

An admin only accessible page to view or edit order and user related entries.

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

SYSTEM ENVIRONMENT

Page | 7

SOFTWARE REQUIREMENTS:

- Eclipse IDE for Enterprise Java Developers - 2019-12
- Oracle- sqldeveloper

Frontend: HTML5

Backend: Springboot- Apache Tomcat server

Accessible using compatible web browsers (Preferably Google Chrome)

HARDWARE REQUIREMENTS:

Minimum hardware requirements:

- Windows OS
- 4GB RAM
- 500GB HDD
- i5 processor 8th gen

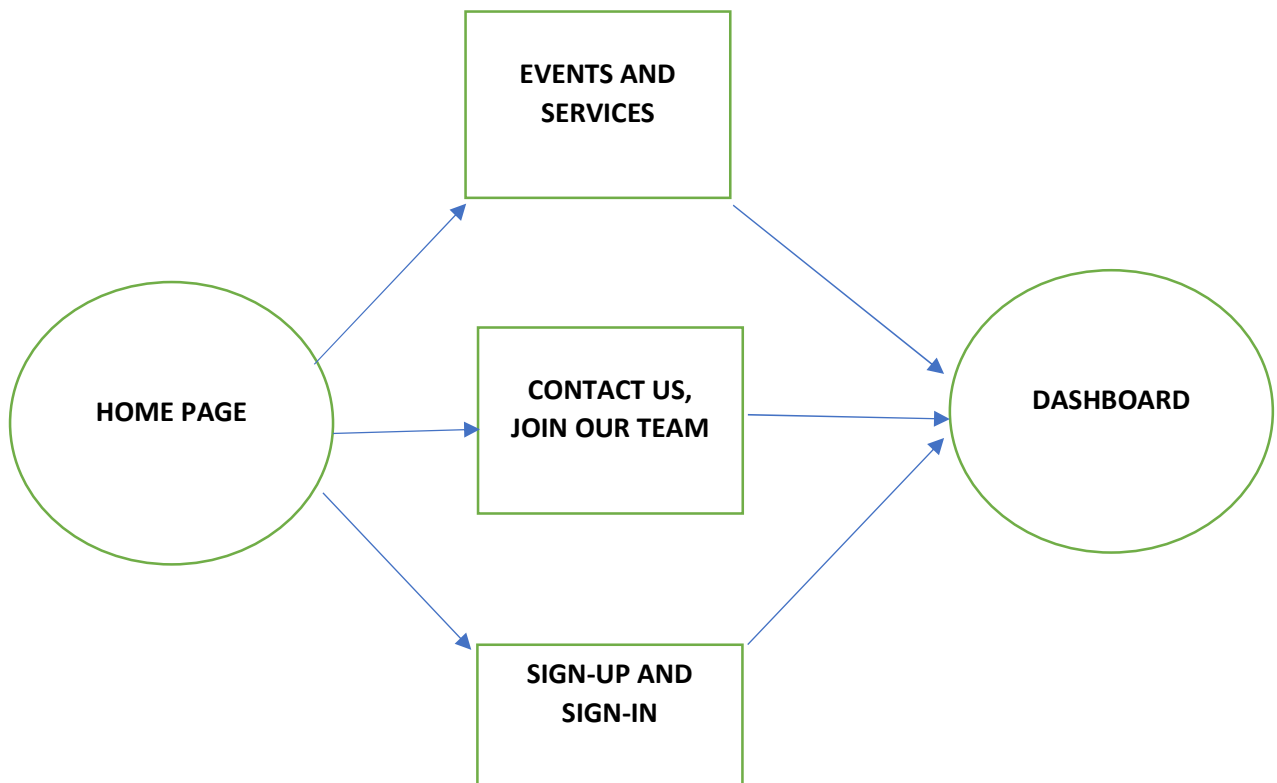
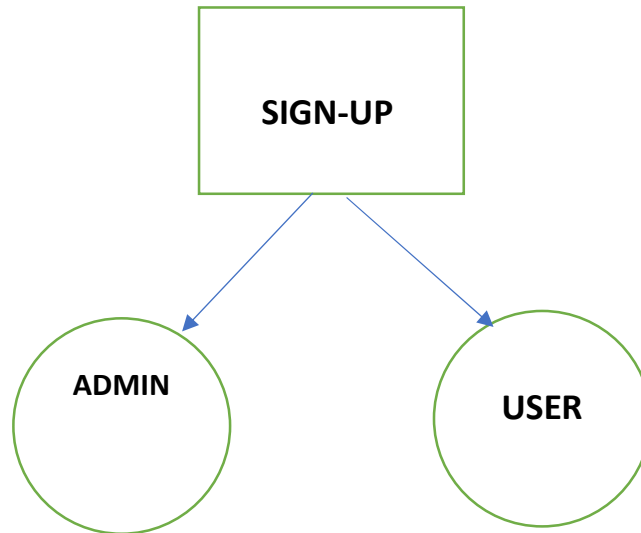
This application is built on a system with Windows 10 OS, 8GB RAM, 1TB HDD and 4GB NVIDIA 1050 Graphics.

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

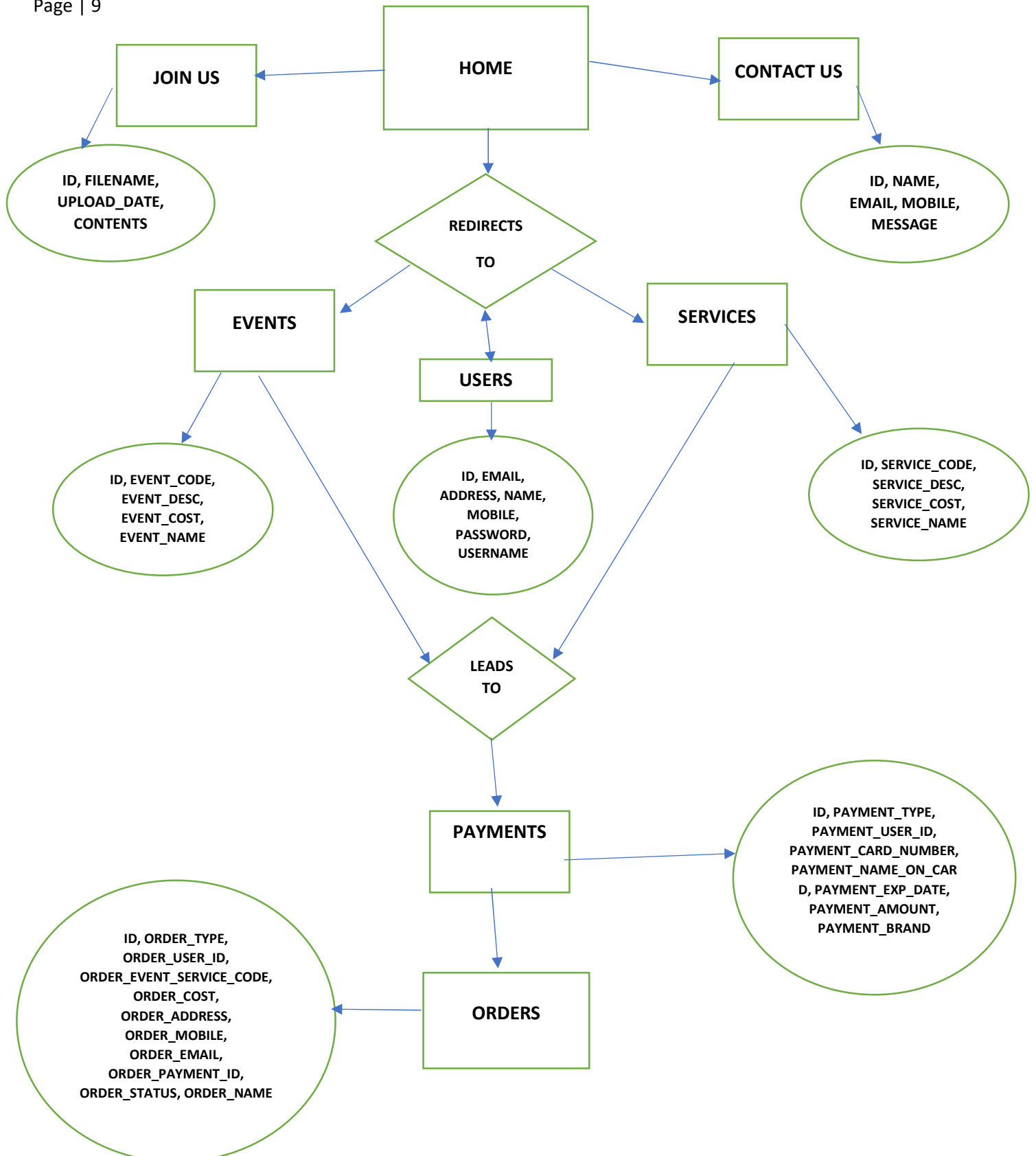
DFD (DATA FLOW DIAGRAM)

Page | 8



ER (ENTITY RELATIONSHIP) DIAGRAM

Page | 9



Surana College

Dept of Computer Sci.

Project- VI Sem BCA - 2020

DATABASE TABLES

ORDERS:

Oracle SQL Developer

File Edit View Navigate Run Team Tools Window Help

Connections

event.sql Welcome Page event ORDERS

Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL

Actions...

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID	NUMBER (38,0)	No	(null)	1 (null)	
2 ORDER_TYPE	VARCHAR2 (20 BYTE)	No	(null)	2 (null)	
3 ORDER_EVENTSERVICE_CODE	VARCHAR2 (20 BYTE)	No	(null)	3 (null)	
4 ORDER_USER_ID	NUMBER	No	(null)	4 (null)	
5 ORDER_COST	NUMBER (38,0)	No	(null)	5 (null)	
6 ORDER_ADDRESS	VARCHAR2 (500 BYTE)	Yes	(null)	6 (null)	
7 ORDER_MOBILE	VARCHAR2 (10 BYTE)	Yes	(null)	7 (null)	
8 ORDER_EMAIL	VARCHAR2 (100 BYTE)	Yes	(null)	8 (null)	
9 ORDER_PAYMENT_ID	NUMBER	Yes	(null)	9 (null)	
10 ORDER_STATUS	VARCHAR2 (20 BYTE)	Yes	(null)	10 (null)	
11 ORDER_CONTACT	VARCHAR2 (20 BYTE)	Yes	(null)	11 (null)	
12 ORDER_DATE	DATE	Yes	(null)	12 (null)	
13 ORDER_CREATED	DATE	Yes	(null)	13 (null)	
14 ORDER_NAME	VARCHAR2 (2000 BYTE)	Yes	(null)	14 (null)	

CONTACT:

Oracle SQL Developer

File Edit View Navigate Run Team Tools Window Help

Connections

event.sql Welcome Page event CONTACT

Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL

Actions...

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID	NUMBER (38,0)	Yes	(null)	1 (null)	
2 NAME	VARCHAR2 (200 BYTE)	Yes	(null)	2 (null)	
3 EMAIL	VARCHAR2 (100 BYTE)	Yes	(null)	3 (null)	
4 MOBILE	VARCHAR2 (10 BYTE)	Yes	(null)	4 (null)	
5 MESSAGE	VARCHAR2 (2000 BYTE)	Yes	(null)	5 (null)	

Surana College

Dept of Computer Sci.

Project- VI Sem BCA - 2020

EVENTS:

Oracle SQL Developer

File Edit View Navigate Run Team Tools Window Help

Connections

event.sql Welcome Page event EVENTS

Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL

Actions...

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID	NUMBER	No	(null)	1 (null)	
2 EVENT_CODE	VARCHAR2 (20 BYTE)	No	(null)	2 (null)	
3 EVENT_DESC	VARCHAR2 (500 BYTE)	No	(null)	3 (null)	
4 EVENT_COST	NUMBER (38, 0)	No	(null)	4 (null)	
5 EVENT_NAME	VARCHAR2 (50 BYTE)	No	(null)	5 (null)	

Page | 11

USERS:

Oracle SQL Developer

File Edit View Navigate Run Team Tools Window Help

Connections

event.sql Welcome Page event USERS

Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL

Actions...

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID	NUMBER (38, 0)	No	(null)	1 (null)	
2 ADDRESS	VARCHAR2 (255 BYTE)	Yes	(null)	2 (null)	
3 EMAIL	VARCHAR2 (255 BYTE)	Yes	(null)	3 (null)	
4 FIRSTNAME	VARCHAR2 (255 BYTE)	Yes	(null)	4 (null)	
5 LASTNAME	VARCHAR2 (255 BYTE)	Yes	(null)	5 (null)	
6 MOBILE	VARCHAR2 (255 BYTE)	Yes	(null)	6 (null)	
7 PASSWORD	VARCHAR2 (255 BYTE)	Yes	(null)	7 (null)	
8 ROLE	VARCHAR2 (255 BYTE)	Yes	(null)	8 (null)	
9 USERNAME	VARCHAR2 (255 BYTE)	Yes	(null)	9 (null)	
10 PICTURE	BLOB	Yes	(null)	10 (null)	
11 PICTURE_BASE64	VARCHAR2 (255 BYTE)	Yes	(null)	11 (null)	

Surana College

Dept of Computer Sci.

Project- VI Sem BCA - 2020

UPLOADS:

Oracle SQL Developer

File Edit View Navigate Run Team Tools Window Help

Connections

Oracle Connections

event

Tables (Filtered)

CONTACT

EVENTS

ORDERS

PAYMENTS

SERVICES

UPLOADS

USERS

Views

Indexes

Packages

Procedures

Functions

event.sql Welcome Page event UPLOADS

Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID	NUMBER(38,0)	Yes	(null)	1 (null)	
2	FILENAME	VARCHAR2(200 BYTE)	Yes	(null)	2 (null)	
3	UPLOAD_DATE	DATE	Yes	(null)	3 (null)	
4	CONTENTS	BLOB	Yes	(null)	4 (null)	

Page | 12

SERVICES:

Oracle SQL Developer

File Edit View Navigate Run Team Tools Window Help

Connections

Oracle Connections

event

Tables (Filtered)

CONTACT

EVENTS

ORDERS

PAYMENTS

SERVICES

UPLOADS

USERS

Views

Indexes

Packages

event.sql Welcome Page event SERVICES

Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes

Actions...

	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	ID	NUMBER	No	(null)	1 (null)	
2	SERVICE_CODE	VARCHAR2(20 BYTE)	No	(null)	2 (null)	
3	SERVICE_NAME	VARCHAR2(50 BYTE)	No	(null)	3 (null)	
4	SERVICE_DESC	VARCHAR2(500 BYTE)	No	(null)	4 (null)	
5	SERVICE_COST	NUMBER(38,0)	No	(null)	5 (null)	

Surana College

Dept of Computer Sci.

Project- VI Sem BCA - 2020

PAYMENTS:

Oracle SQL Developer

File Edit View Navigate Run Team Tools Window Help

Connections

event.sql Welcome Page event PAYMENTS

Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL

Actions...

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID	NUMBER	No	(null)	1 (null)	
2 PAYMENT_TYPE	VARCHAR2(20 BYTE)	Yes	(null)	2 (null)	
3 PAYMENT_USER_ID	NUMBER	Yes	(null)	3 (null)	
4 PAYMENT_CARDNUMBER	VARCHAR2(20 BYTE)	Yes	(null)	4 (null)	
5 PAYMENT_NAME_ON_THE_CARD	VARCHAR2(100 BYTE)	Yes	(null)	5 (null)	
6 PAYMENT_EXP_MONTH	VARCHAR2(20 BYTE)	Yes	(null)	6 (null)	
7 PAYMENT_EXP_YEAR	VARCHAR2(20 BYTE)	Yes	(null)	7 (null)	
8 PAYMENT_CVV	VARCHAR2(20 BYTE)	Yes	(null)	8 (null)	
9 PAYMENT_AMOUNT	NUMBER(38,0)	Yes	(null)	9 (null)	
10 PAYMENT_BRAND	VARCHAR2(255 BYTE)	Yes	(null)	10 (null)	

Page | 13

SCREENSHOTS

Sign-in:

Elegant Events

Home Events Services Gallery About Us Contact Us Join Our Team Sign In New User? Sign Up

What are you looking for? Search

Sign In Here

Enter Email or Mobile

Password

Sign In

[Forgot Your Password? Click here to Reset](#)

[New User? Sign Up](#)

[Continue to Explore](#)

© Copyrights Elegant Events 2020

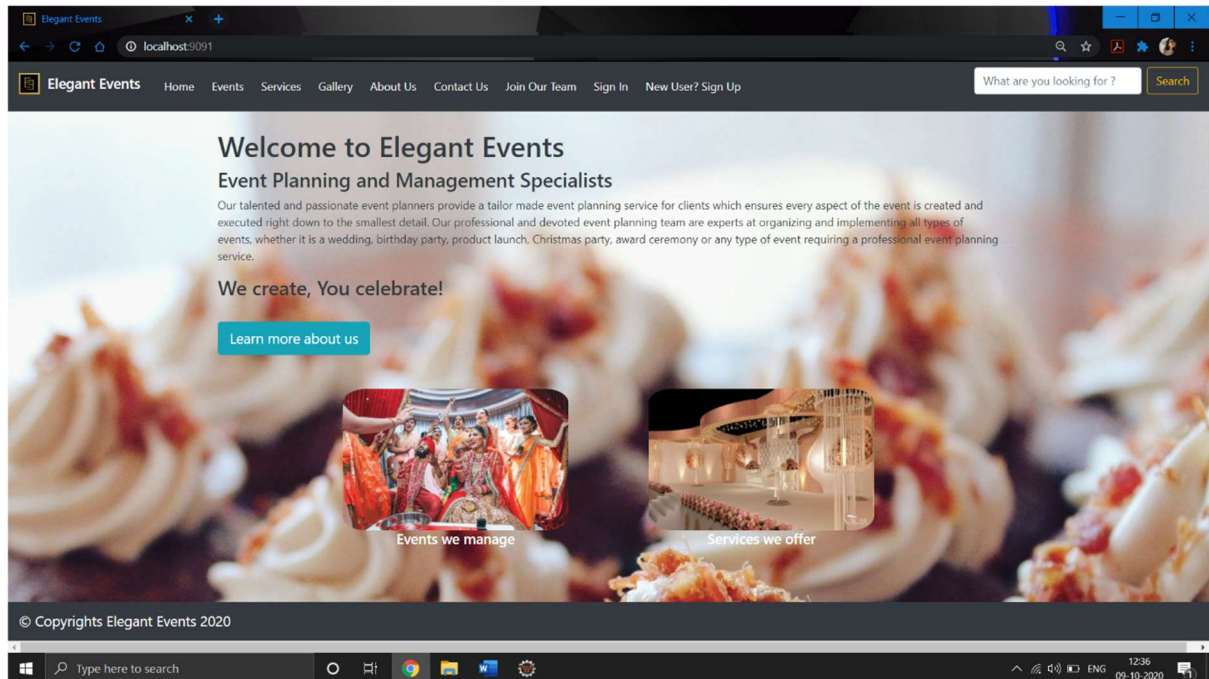
Surana College

Dept of Computer Sci.

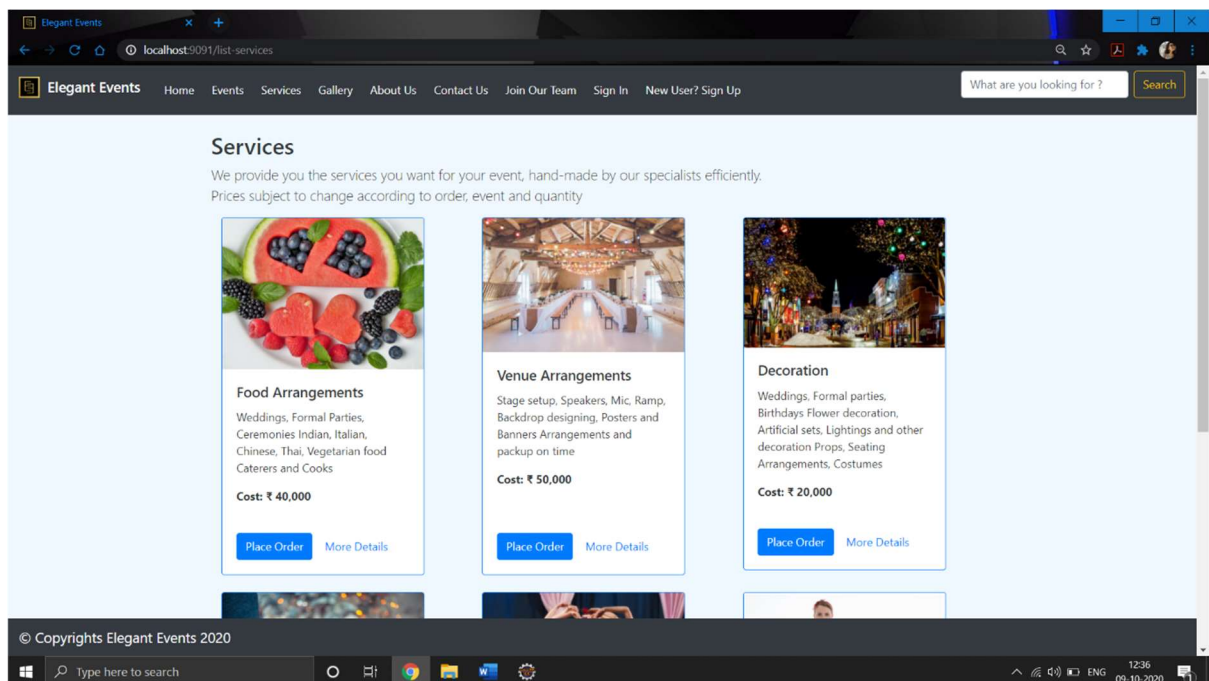
Project- VI Sem BCA - 2020

Home-page:

Page | 14



Services-page:



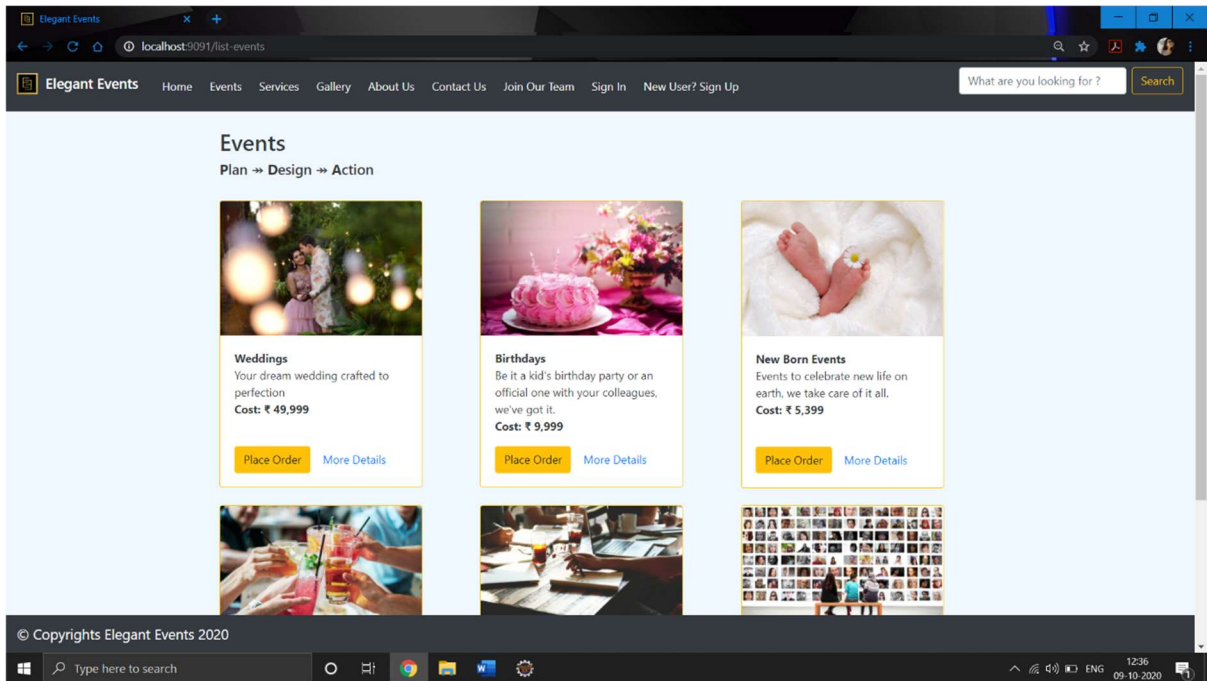
Surana College

Dept of Computer Sci.

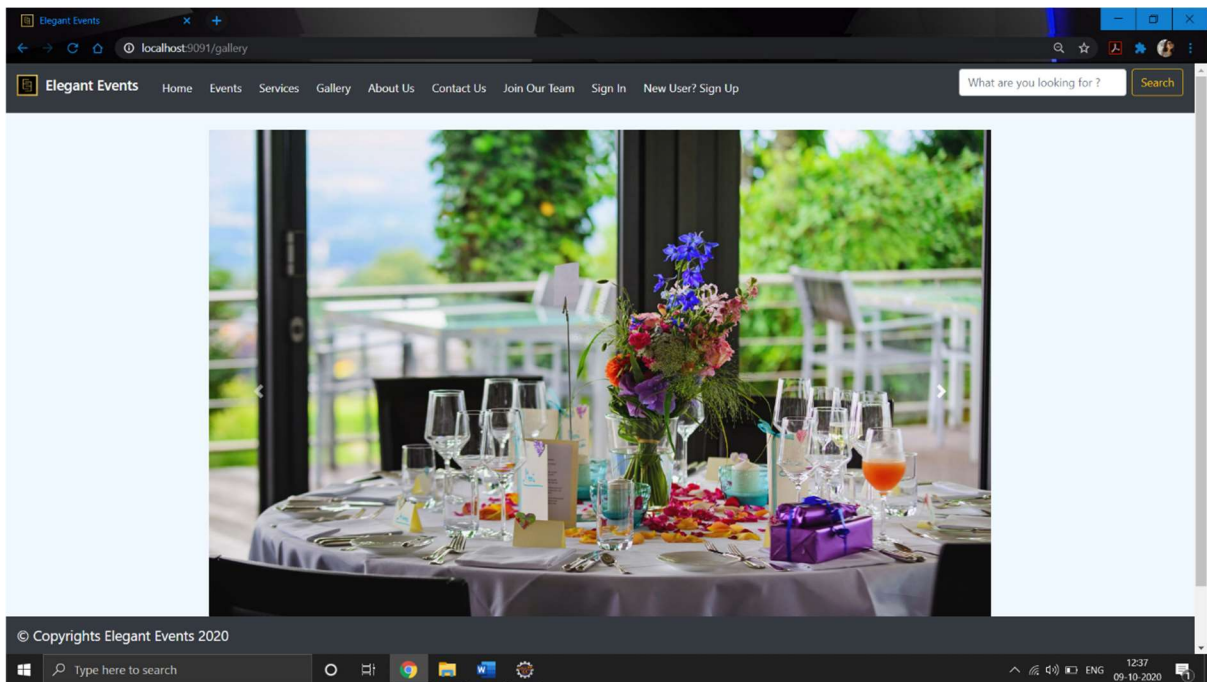
Project- VI Sem BCA - 2020

Events-page:

Page | 15



Gallery:



Surana College

Dept of Computer Sci.

Project- VI Sem BCA - 2020

Contact-Us page:

Page | 16

The screenshot shows a web browser window displaying the 'Contact Us' page of 'Elegant Events'. The browser's address bar shows 'localhost:9091/contact'. The website's navigation bar includes links for Home, Events, Services, Gallery, About Us, Contact Us, Join Our Team, Sign In, and New User? Sign Up. A search bar is located on the right side of the navigation bar. The main content area has a light blue background and contains the following elements:

- Contact Us**: A heading followed by the text 'Contact us for any queries. Write to us and we will get back to you with the best.'
- Full Name**: A text input field with the placeholder 'Your Name'.
- Email Address**: A text input field with the placeholder 'Email'.
- Mobile**: A text input field with the placeholder 'Mobile'.
- Message**: A large text area with the placeholder 'Enter your message'.
- Send Message**: A blue button.

The footer of the page displays '© Copyrights Elegant Events 2020'.

About-us:

The screenshot shows a web browser window displaying the 'About Us' page of 'Elegant Events'. The browser's address bar shows 'localhost:9091/about'. The website's navigation bar is identical to the Contact Us page. The main content area has a light blue background and contains the following elements:

- About Us**: A heading followed by the text 'Elegant Events, a 2020 company is built to serve its customers.'
- We believe in understanding the emotions and the professional ethics behind every project we look into.**: A bold statement followed by the text 'The company was found to serve people and make their events the best.'
- Elegant Events is not only a customer friendly service station but also a platform for many event managers to grow.**: A bold statement.
- Our Motto**: A heading followed by a bulleted list:
 - Reduce, Reuse And Recycle : Our team believes in providing Eco friendly services and reduces the usage of plastic in events.
 - Motivate : Our team believes in the work and motivate others to work attitude. Here in the team, every voice is listened to.
 - Love : Every service that the team provides for you is tailored to perfection with dedication and love

The footer of the page displays '© Copyrights Elegant Events 2020'.

Surana College

Dept of Computer Sci.

Project- VI Sem BCA - 2020

SOURCE CODE

FRONT-END:

Home page:

Page | 17

```
<%@ include file="common/header.jspf"%>
<%@ include file="common/navigation.jspf"%>

<style>
img {
border: 0px solid white;
}
img:hover {
border: 4px solid blue;
}
</style>

<div style="height:90%; background-image: url(/images/wedding.jpg); background-
repeat:no-repeat;">
<div class="container">
<!-- <h1 class="display-4">Hello ${name}</h1> -->
<br>
<h1>Welcome to Elegant Events</h1>
<!-- <hr class="my-4"> -->
<h3>Event Planning and Management Specialists</h3>
<p>
Our talented and passionate event planners provide a tailor made
event planning service for clients which ensures every aspect of the
event is created and executed right down to the smallest detail. Our
professional and devoted event planning team are experts at organizing
and implementing all types of events, whether it is a wedding,
birthday party, product launch, Christmas party, award ceremony or any
type of event requiring a professional event planning service.
</p>

<h3>We create, You celebrate!</h3>
<br>
<a class="btn btn-info btn-lg" href="about" role="button">Learn more about us</a>
<br><br>

<br>

</div>

<div class="row">

<div class="col">
</div>

<div class="col">
```

Surana College

Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
<h5 style="text-align: center;">
<a href="list-events" class="text-white">

<br>
Events we manage
</a>
</h5>
</div>

<div class="col">
<h5 style="text-align: center;">
<a href="list-events" class="text-white">

<br>
Services we offer
</a>
</h5>
</div>

<div class="col">
</div>

</div>

</div>
<%@ include file="common/footer.jspf"%>
```

About Us:

```
<%@ include file="common/header.jspf"%>
<%@ include file="common/navigation.jspf"%>

<div class="jumbotron jumbotron-fluid">
<div class="container">
<h2>About Us</h2>
<p class="lead">Elegant Events, a 2020 company is built to serve its
customers.</p>
</div>

<div class="container">
<h3> We believe in understanding the emotions and the professional ethics behind
every project we look into.</h3>
<p class="lead">The company was found to serve people and make their events the
best. </p>
<p class="lead"> Elegant Events is not only a customer friendly service station
but also a platform for many event managers to grow. </p>
</div>
</div>

<div class="container">
<h3 class="display-5"> Our Motto </h3>
<ul>
<li><h3 class="lead">Reduce, Reuse And Recycle: Our team believes in providing Eco
friendly services and reduces the usage of plastic in events. </h3> </li>
```

Surana College

Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
<li><h3 class="Lead">Motivate : Our team believes in the work and motivate others
to work attitude. Here in the team, every voice is listened to.</h3></li>
<li><h3 class="Lead">Love : Every service that the team provides for you is
tailored to perfection with dedication and love</h3></li>
</ul>
</div>
<br>
<%@ include file="common/footer.jspf"%>
```

Page | 19

Contact us:

```
<%@ include file="common/header.jspf"%>
<%@ include file="common/navigation.jspf"%>

<br>
<div class="container">
<!-- contact section -->
<section id="contact" class="section">
<div class="container">
<div class="section-header">
<h2 class="wow fadeInDown animated">Contact Us</h2>
<p class="Lead">
Contact us for any queries. Write to us and we will get back to you with the best.
</p>
</div>

<form:form method="post" commandName="contact">
<div class="form-group">
<label for="name">Full Name</label>
<input type="text" class="form-control" id="name" name="name" placeholder="Your
Name">
</div>

<div class="row">
<div class="col">
<div class="form-group">
<label for="email">Email Address</label>
<input type="email" class="form-control" id="email" name="email"
placeholder="Email">
</div>
</div>
<div class="col">
<div class="form-group">
<label for="mobile">Mobile</label>
<input type="text" class="form-control" id="mobile" name="mobile"
placeholder="Mobile">
</div>
</div>
</div>

<div class="form-group">
<label for="message">Message</label>
<textarea class="form-control" id="message" name="message" rows="3"
placeholder="Enter your message"></textarea>
</div>
<div class="form-group">
```

Surana College

Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
<button type="submit" class="btn btn-primary" id="send">Send Message</button>
</div>
</form:form>
```

```
<br>
```

Page | 20

```
<c:if test="${not empty message and messagetype == 'error'}">
<div style="color: red">
<h5>${message}</h5>
</div>
</c:if>

<c:if test="${not empty message and messagetype == 'success'}">
<div style="color: green">
<h5>${message}</h5>
</div>
</c:if>

</div>
</section>
<!-- contact section -->
</div>

<%@ include file="common/footer.jspf"%>
```

Events:

```
<%@ include file="common/header.jspf" %>
<%@ include file="common/navigation.jspf" %>
<div class="container">
<form:form method="post" commandName="event">
<form:hidden path="id" />

<fieldset class="form-group">
<form:label path="name">Event name</form:label>
<form:input path="name" type="text" class="form-control"
required="required" />
<form:errors path="name" cssClass="text-warning" />
</fieldset>

<fieldset class="form-group">
<form:label path="desc">Description</form:label>
<form:input path="desc" type="text" class="form-control"
required="required" />
<form:errors path="desc" cssClass="text-warning" />
</fieldset>

<fieldset class="form-group">
<form:label path="category">Category</form:label>
<form:input path="category" type="text" class="form-control"
required="required" />
<form:errors path="category" cssClass="text-warning" />
</fieldset>

<fieldset class="form-group">
<form:label path="cost">Cost</form:label>
```

Surana College

Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
<form:input path="cost" type="text" class="form-control"
required="required" />
<form:errors path="cost" cssClass="text-warning" />
</fieldset>

<fieldset class="form-group">
<form:label path="date">Event Date</form:label>
<form:input path="date" type="text" class="form-control"
required="required" />
<form:errors path="date" cssClass="text-warning" />
</fieldset>

<fieldset class="form-group">
<form:label path="address">Address</form:label>
<form:input path="address" type="text" class="form-control"
required="required" />
<form:errors path="address" cssClass="text-warning" />
</fieldset>

<button type="submit" class="btn btn-success">Save</button>
<a href="list-events">Back</a>
</form:form>
</div>
<%@ include file="common/footer.jspf" %>
```

Event details:

```
<%@ include file="common/header.jspf"%>
<%@ include file="common/navigation.jspf"%>
<br>
<div class="container">
<!-- our team section -->
<section id="teams" class="section teams">
<div class="section-header">
<h2 class="wow fadeInDown animated">Event Details</h2>
<p class="lead">
Choose your event <br>
</p>
</div>
<div class="container">
<div class="person" id="weddings">

<div class="person-content">
<br>
<h3>Weddings</h3>
<br>
<h5 class="role">Opt your dream wedding</h5>
<p>We design, plan and manage your wedding in a simple and
organized manner. We work with you every step of the way to ensure
your special day is a dream come true. Our experienced team makes
sure your experience before, during and after the wedding is
everything and more than you imagined.</p>
<b> </b>
<h4>What we provide</h4>
<b> Elements </b>
```

Surana College

Dept of Computer Sci.

Project- VI Sem BCA - 2020

<p>Wedding Stationery, Catering, Wedding Cake, Hospitality/Travel, Venue Selection, Decor, Entertainment + Sound/Light, Photo and Video, Makeup and Styling, Gifts/Giveaways, Customized Honeymoons, Post Wedding Party</p>

<p>

 8 Elements per package

</p>

<p>

Add ons Cost: ₹ 5,000 per element

</p>

Place Order

</div>

<div class="container">

<div class="person" id="birthdays">

<div class="person-content">

<h3>Birthdays</h3>

<h5 class="role">We give you the best birthdays to celebrate</h5>

<p>Planning a birthday party in our busy city is not an easy job, it takes skills to make it look so easy and beautiful</p>

<h4>What we provide</h4>

 Kids party

<p>Balloons and decoration, customized cake, Themed party, Return gifts, Fun activities for kids, Food arrangements, Magicians and other performers, Games, Venue arrangements</p>

<p>

 5 elements per purchase

</p>

 Formal party

<p><u>Venue</u> arrangements, Emcees, Customized cakes, Return gifts, Performers, Food and drinks, Fun activities.</p>

<p>

 4 elements per purchase

</p>

<p>

Add ons Cost: ₹ 5,000 per element

</p>

Place Order

</div>

<div class="person" id="newbornevents">

<div class="person-content">

<h3>New Born Events</h3>

Surana College

Dept of Computer Sci.

Project- VI Sem BCA - 2020

Page | 23

```
<br>
<h5 class="role">Child-birth is a man-made miracle.</h5>
<p>We understand your care and love towards your baby and
create beautiful events to celebrate the new life on earth.</p>
<b> </b>
<h4>What we provide</h4>
<b> Events </b>
<p>Baby showers, New born party, Naming ceremony, Parties to
celebrate all the firsts, Baby dress up photoshoot, Baby
timeline, 1st year birthday and other religious ceremonies.</p>
<p>
<b> 1 event per purchase</b>
</p>
<b></b> <b> Prices subject to change according to event*</b>
<p>
<b>Add ons Cost: </b><span>&#8377;</span> 10,000 per event
</p>
<a href="pPlaceorder?code=NEWBORN&type=Event" class="btn btn-warning">Place
Order</a> <br>

<br>
</div>
<div class="person" id="parties">

<div class="person-content">
<br>
<h3>Parties</h3>
<br>
<h5 class="role">All types of parties, the fun and the
formal way.</h5>
<p>A party, a grand one with a lot of people and music or
maybe even a small get-together.</p>
<b> </b>
<h4>What we provide</h4>
<b> Elements </b>
<p>&u>Venue arrangements, music, food and drinks, DJ, Live
concerts, Artists like stand up comedians, singers, dancers,
Emcees</p>
<p>
<b> 3 elements per purchase</b>
</p>
<b> </b>
<p>
<b>Add ons Cost: </b><span>&#8377;</span> 4,999 per element
</p>
<a href="pPlaceorder?code=PARTIES&type=Event" class="btn btn-warning">Place
Order</a>

</div>
<br> <br>
</div>
<div class="person" id="corpevents">

<div class="person-content">
```

Project- VI Sem BCA - 2020

```
<br>
<h3>Corporate events</h3>
<br>
<h5 class="role">We design the most elegant formal events while we let you handle the business.</h5>
<p>Any business meeting or client call, you sit back and relax while we provide you with the best.</p>
<b> </b>
<h4>What we provide</h4>
<b> Elements </b>
<p>Venue arrangements, Decor, Mic, Projectors, Seating arrangement and a Formal approach to any of your corporate events like Board meetings, Award ceremonies, Product launch or even Office parties.</p>
<p>
<b> 3 elements per purchase</b>
</p>
<b> </b>
<p>
<b>Add ons Cost: </b><span>₹8377;</span> 6,999 per element
</p>
<a href="placeorder?code=Corp&type=Event" class="btn btn-warning">Place Order</a>
</div>
<br> <br>
</div>
<div class="person" id="socialservice">

<div class="person-content">
<br>
<h3>Social service</h3>
<br>
<h5 class="role">
<b> We partner with you to sponsor any fund-raising events to help the society.</b>
</h5>

<b> If you've got any idea for a deed, leave it to us and we will create an event out of it</b>
<br>
<a href="contact">Contact us for more information</a>
<br> <br> <br> <br>
</div>
<br>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</section>

</div>
<%@ include file="common/footer.jspf"%>
```


Surana College

Dept of Computer Sci.

Project- VI Sem BCA - 2020

Services:

Page | 25

```
<%@ include file="common/header.jspf" %>
<%@ include file="common/navigation.jspf" %>
<div class="container">
<form:form method="post" commandName="service">
<form:hidden path="id" />
<fieldset class="form-group">
<form:label path="desc">Description</form:label>
<form:input path="desc" type="text" class="form-control"
required="required" />
<form:errors path="desc" cssClass="text-warning" />
</fieldset>

<fieldset class="form-group">
<form:label path="category">Category</form:label>
<form:input path="category" type="text" class="form-control"
required="required" />
<form:errors path="category" cssClass="text-warning" />
</fieldset>

<fieldset class="form-group">
<form:label path="cost">Cost</form:label>
<form:input path="cost" type="text" class="form-control"
required="required" />
<form:errors path="cost" cssClass="text-warning" />
</fieldset>

<button type="submit" class="btn btn-success">Save</button>
<a href="list-services">Back</a>
</form:form>
</div>
<%@ include file="common/footer.jspf" %>
```

Service details:

```
<%@ include file="common/header.jspf"%>
<%@ include file="common/navigation.jspf"%>
<br>
<div class="container">
<!-- our team section -->
<section id="teams" class="section teams">
<div class="section-header">
<h2 class="wow fadeInDown animated">Service Details</h2>
<p class="Lead">
Choose your Service <br>
</p>
</div>
<div class="container">
```

Surana College

Dept of Computer Sci.

Project- VI Sem BCA - 2020

Page | 26

```
<div class="person" id="food">

<div class="person-content">
<br>
<h3>Food Arrangements</h3>
<br>
<h5 class="role">Take your taste buds on an adventure with a
large variety of food in so many different cuisines.</h5>
<p>Choose your cuisine and event, Sit back and relax as we
provide you with the most delicious and hygienic food</p>
<b> </b>
<h4>What we provide</h4>
<b> Cuisine </b>
<p>Weddings, Formal Parties, Ceremonies Indian, Italian,
Chinese, Thai, Vegetarian food Caterers and Cooks</p>
<p>
<b> 1 Elements per package</b>
</p>
<p>Pick your event and cuisine accordingly. Prices are subject
to change depending on quantity and event.</p>
<p>
<b>Add ons Cost: </b><span>&#8377;</span> 25,000 per element
</p>
<a href="pPlaceorder?code=FOOD&type=Service" class="btn btn-warning">Place
Order</a> <br>
</div>
</div>

<br>

<div class="person" id="venue">

<div class="person-content">
<br>
<h3>Venue Arrangements</h3>
<br>
<h5 class="role">On the spot arrangements required for any
event will be delivered to you on time.</h5>
<p>Pick all the spot requirements and facilities, while we
tailor them to perfection for you.</p>
<b> </b>
<h4>What we provide</h4>
<b> Arrangements </b>
<p>Stage setup, Speakers, Mic, Ramp, Backdrop designing,
Posters and Banners. Arrangements and packup on time</p>
<p>
<b> 1 Element per package</b>
</p>
<p>Pick your event and facility accordingly. Prices are subject
to change depending on quantity and event.</p>
<p>
<b>Add ons Cost: </b><span>&#8377;</span> 8,999 per element
</p>
```

Surana College

Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
<a href="placeorder?code=VENUE&type=Service" class="btn btn-warning">Place  
Order</a> <br>  
<br>  
</div>
```

Page | 27

```
<div class="person" id="decor">  
  
<div class="person-content">  
<br>  
<h3>Decor</h3>  
<br>  
<h5 class="role">On the spot arrangements required for any  
event will be delivered to you on time.</h5>  
<p>Our specialists at design have beautiful decoration  
handmade just for you.</p>  
<b> </b>  
<h4>What we provide</h4>  
<b> Elements </b>  
<p>Flower decoration, Artificial sets, Lightings and make-up.  
Props, Seating Arrangements, Costumes</p>  
<p>  
<b> 2 Elements per package</b>  
</p>  
<p>Pick your event and decor accordingly. Prices are subject  
to change depending on quantity and event.</p>  
<p>  
<b>Add ons Cost: </b><span>&#8377;</span> 2,999 per element  
</p>  
<a href="placeorder?code=DECOR&type=Service" class="btn btn-warning">Place  
Order</a> <br>  
<br>  
</div>
```

```
<div class="person" id="gift">  
  
<div class="person-content">  
<br>  
<h3>Gifting</h3>  
<br>  
<h5 class="role">Handmade and antique gifts, well packed  
that defines your legacy.</h5>  
<p>Our specialists at design have beautiful gifts, hand-made  
just for you.</p>  
<b> </b>  
<h4>What we provide</h4>  
<b> Elements </b>  
<p>Handmade gifts, Formal souvenirs, Invitation and Thank you  
cards</p>  
<p>  
<b> 2 Elements per package</b>  
</p>  
<p>Prices are subject to change depending on quantity and  
event.</p>  
<p>
```

Surana College

Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
<b>Add ons Cost: </b><span>&#8377;</span> 3,999 per element
</p>
<a href="pLaceorder?code=GIFT&type=Service" class="btn btn-warning">Place
Order</a> <br>
<br>
</div>
```

Page | 28

```
<div class="person" id="perform">

<div class="person-content">
<br>
<h3>Performers</h3>
<br>
<h5 class="role">On the spot arrangements required for any
event will be delivered to you on time.(Subject to <u>availability</u>)
</h5>
<p>We get you in touch with the best of artists worldwide,
right for your event and budget.Respecting the artist is what
we ask in return.</p>
<b> </b>
<h4>What we provide</h4>
<b> Artists </b>
<p>Live performances- Magic shows, Dance shows, Event
hosting. Performer artists- Dancers, Magicians, <u>Choreographers</u>
</p>
<p>
<b> 1 Artist/crew per package</b>
</p>
<p>Event date and time to be specified accurately. Prices
might change subject to availability.</p>
<p>
<b>Add ons Cost: </b><span>&#8377;</span> 35,999 per crew
</p>
<a href="pLaceorder?code=PERF&type=Service" class="btn btn-warning">Place
Order</a> <br>
<br>
</div>
```

```
<div class="person" id="man">

<div class="person-content">
<br>
<h3>Man power</h3>
<br>
<h5 class="role">We provide helpers to make your work
easy. Be it cleaning, serving or managing.</h5>
<p>Our specialists have just the right people at your
service.</p>
<b> </b>
<h4>What we provide</h4>
<b> Services </b>
<p>Helpers for events,Cleaning, Cooking, Dish washing, Food
serving, Interior designers,Gardeners, Valets</p>
```

Project- VI Sem BCA - 2020

Page | 29

```
<%@ include file="common/header.jspf"%>
<%@ include file="common/navigation.jspf"%>

<br>
<!--
<div class="container">
<section id="contact" class="section">
<div class="container">
<div class="section-header">
<h2 class="wow fadeInDown animated">Gallery</h2>
<h4>Few beautiful memories to remember us by.</h4>
</div>
</div>
</section>
</div> -->
```

```
<!--Carousel Wrapper-->
<div id="gallery-carousel" class="container carousel slide carousel-fade" data-ride="carousel">
<!--Indicators-->
<ol class="carousel-indicators">
<li data-target="#gallery-carousel" data-slide-to="0" class="active"></li>
<li data-target="#gallery-carousel" data-slide-to="1"></li>
<li data-target="#gallery-carousel" data-slide-to="2"></li>
<li data-target="#gallery-carousel" data-slide-to="3"></li>
</ol>
<!--/.Indicators-->
<!--Slides-->
<div class="carousel-inner" role="listbox">
```

Surana College

Dept of Computer Sci.

Project- VI Sem BCA - 2020

Page | 30

```
<div class="carousel-item active">

</div>

<div class="carousel-item">

</div>

<div class="carousel-item">

</div>

<div class="carousel-item">
<video class="video-fluid" autoplay loop muted>
<source src="https://mdbootstrap.com/img/video/Tropical.mp4" type="video/mp4" />
</video>
</div>

<div class="carousel-item">

</div>

<div class="carousel-item">

</div>

<div class="carousel-item">

</div>

<div class="carousel-item">
<video class="video-fluid" autoplay loop muted>
<source src="https://mdbootstrap.com/img/video/forest.mp4" type="video/mp4" />
</video>
</div>

<div class="carousel-item">

</div>

<div class="carousel-item">

</div>

<div class="carousel-item">

</div>

<div class="carousel-item">
<video class="video-fluid" autoplay loop muted>
<source src="https://mdbootstrap.com/img/video/Agua-natural.mp4" type="video/mp4" />
</video>
</div>
```

Surana College

Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
<div class="carousel-item">

</div>

</div>
<!--/.Slides-->
<!--Controls-->
<a class="carousel-control-prev" href="#gallery-carousel" role="button" data-
slide="prev">
<span class="carousel-control-prev-icon" aria-hidden="true"></span>
<span class="sr-only">Previous</span>
</a>
<a class="carousel-control-next" href="#gallery-carousel" role="button" data-
slide="next">
<span class="carousel-control-next-icon" aria-hidden="true"></span>
<span class="sr-only">Next</span>
</a>
<!--/.Controls-->
</div>
<!--Carousel Wrapper-->

<br><br><br><br>
```

```
<%@ include file="common/footer.jspf"%>
```

Payments:

```
<%@ include file="common/header.jspf" %>
<%@ include file="common/navigation.jspf" %>
<br>
<div class="container">
<h3>Payments</h3>
</div>
<br>
<div class="container">
<c:if test="${not empty payments}">
<table class="table table-striped">
<thead>
<tr>
<th>Payment #</th>
<th>Payment Brand</th>
<th>Payment Type</th>
<th>Name on the Card</th>
<th>Payment Amount</th>
<th>User #</th>
<th></th>
</tr>
</thead>
<tbody>
<c:forEach items="${payments}" var="payment">
<tr>
<td>${payment.id}</td>
<td>${payment.brand}</td>
<td>${payment.type}</td>
<td>${payment.nameOnTheCard}</td>
```

Surana College

Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
<td>₹ ${payment.amount}</td>
<td>${payment.userId}</td>
<td><a type="button" class="btn btn-link" href="#">More Details</a></td>
</tr>
</c:forEach>
</tbody>
</table>
</c:if>

<c:if test="${empty payments}">
<div style="color: green">
<h5>There are no Payments.</h5>
</div>
</c:if>

<br>
<a href = "dashboard" class="btn btn-link">Back to Dashboard</a>
</div>

<br><br>

<%@ include file="common/footer.jspf" %>
```

My account:

```
<%@ include file="common/header.jspf" %>
<%@ include file="common/navigation.jspf" %>

<br>
<div class="container">
<h3>User Profile</h3>
<br>

<div class="row">
<div class="col-4">
<form:form method="post" action="uploadpic" enctype="multipart/form-data">

<div class="container text-center">
<c:if test="${not empty profilepic}">

</c:if>
<c:if test="${empty profilepic}">

</c:if>
<br>
<div class="file btn btn-link" style="position: relative; overflow: hidden;">
Change Picture
<input type="file" name="fileprofile" id="fileprofile" accept="image/*"
style="position: absolute; font-size: 50px; opacity: 0; right: 0; top: 0;" />
</div>
<br>
<button type="submit" class="btn btn-warning" id="uploadimage" style="text-align:
center">Update Picture</button>
```


Surana College

Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
</div>

<!-- <div class="card bg-light text-center" style="width: 19rem;">
<c:if test="${not empty profilepic}">

</c:if>
<c:if test="${empty profilepic}">

</c:if>
<div class="file btn btn-link" style="position: relative; overflow: hidden;">
Change Picture
<input type="file" name="fileprofile" id="fileprofile" accept="image/*"
style="position: absolute; font-size: 50px; opacity: 0; right: 0; top: 0;" />
</div>
<button type="submit" class="btn btn-warning" id="uploadimage" style="text-align:
center">Update Picture</button>
</div> --%>

<script lang="JavaScript">
document.getElementById("fileprofile").addEventListener(
"change", changeProfilePic, false);
function changeProfilePic() {
var file = this.files[0];
if (file.type.startsWith('image/')) {

const img = document.getElementById("profile_pic");
img.file = file;
const reader = new FileReader();
reader.onload = (function(aImg) {
return function(e) {
aImg.src = e.target.result;
};
})(img);
reader.readAsDataURL(file);
}
}
</script>
</form:form>
</div>
<div class="col-8">

<form:form method="post" commandName="myaccount">
<div class="row">
<div class="col">
<label for="firstname">First Name</label>
<input type="text" id="firstName" name="firstName" class="form-control"
placeholder="First name" value="${user.firstName}" required>
</div>
<div class="col">
<label for="lastname">Last Name</label>
<input type="text" id="lastName" name="lastName" class="form-control"
placeholder="Last name" value="${user.lastName}" required>
</div>
</div>
</div>
```

Surana College

Dept of Computer Sci.

Project- VI Sem BCA - 2020

Page | 34

```
<br>
<div class="row">
<div class="col">
<label for="email">Email</label>
<input type="email" id="email" name="email" class="form-control"
placeholder="Email" value="{user.email}" required>
</div>
<div class="col">
<label for="mobile">Mobile</label>
<input type="text" id="mobile" name="mobile" class="form-control"
placeholder="Mobile" value="{user.mobile}" required>
</div>
</div>
<br>
<div class="form-group">
<label for="address">Address</label>
<textarea class="form-control" id="address" name="address" rows="3"
required>{user.address}</textarea>
</div>
<div class="form-group">
<button type="submit" class="btn btn-success btn-lg" id="update">Update
Details</button>
</div>
</form:form>
</div>
</div>
<br>
<div class="row">

<div class="col-4"></div>

<div class="col-8">
<c:if test="{not empty message and messagetype == 'error'}">
<div style="color: red">
<h5>{message}</h5>
</div>
</c:if>

<c:if test="{not empty message and messagetype == 'success'}">
<div style="color: green">
<h5>{message}</h5>
</div>
</c:if>
</div>

</div>

<%@ include file="common/footer.jspf" %>
```

My orders:

```
<%@ include file="common/header.jspf" %>
<%@ include file="common/navigation.jspf" %>
```

Surana College

Dept of Computer Sci.

Project- VI Sem BCA - 2020

Page | 35

```
<br>
<div class="container">
<h3>My Orders</h3>
</div>
<div class="container">
<c:if test="${not empty myorders}">
<table class="table table-striped">
<thead>
<tr>
<th>Order #</th>
<th>Type</th>
<th>Details</th>
<th>Date</th>
<th>Status</th>
<!-- <th></th> -->
<th></th>
</tr>
</thead>
<tbody>
<c:forEach items="${myorders}" var="order">
<tr>
<td style="width: 100px !important;">${order.id}</td>
<td style="width: 100px !important;">${order.type}</td>
<td style="width: 400px !important;">${order.name}</td>
<td style="width: 100px !important;">${order.orderDate}</td>
<td style="width: 50px !important;">${order.status}</td>
<!-- <td style="width: 100px !important;"><a type="button" class="btn btn-link"
href="#">Cancel Order</a></td> -->
<td style="width: 100px !important;"><a type="button" class="btn btn-link"
href="orderdetails?id=${order.id}">Order Details</a></td>
</tr>
</c:forEach>
</tbody>
</table>
</c:if>

<c:if test="${empty myorders}">
<div style="color: green">
<h5>There are no orders.</h5>
</div>
</c:if>
<br>
<a href="/" class="btn btn-link">Continue to Explore</a>
</div>

<br><br><br>
```

```
<%@ include file="common/footer.jspf" %>
```

Dashboard:

```
<%@ include file="common/header.jspf" %>
<%@ include file="common/navigation.jspf" %>
<br>
```

Surana College

Dept of Computer Sci.

Project- VI Sem BCA - 2020

Page | 36

```
<div class="container">
<h3>Dashboard</h3>
</div>
<div class="container">
<div class="row">
<div class="col-sm-4">
<div class="card border-secondary">
<div class="card-header"><h5>Orders<span class="badge badge-
primary">${number_orders}</span></h5></div>
<div class="card-body">
<p class="card-text">View processed and upcoming orders in queue.</p>
<a href="orders" class="btn btn-outline-primary">View Orders</a>
</div>
</div>
</div>
<div class="col-sm-4">
<div class="card border-secondary">
<div class="card-header"><h5>Payments<span class="badge badge-
success">${number_payments}</span></h5></div>
<div class="card-body">
<p class="card-text">Shows payment details made for orders placed by
customers.</p>
<a href="payments" class="btn btn-outline-success">View Payments</a>
</div>
</div>
</div>
<div class="col-sm-4">
<div class="card border-secondary">
<div class="card-header"><h5>Messages<span class="badge badge-
danger">${number_messages}</span></h5></div>
<div class="card-body">
<p class="card-text">View messages and feedback from users.</p>
<br>
<a href="messages" class="btn btn-outline-danger">View Messages</a>
</div>
</div>
</div>
</div>
<br>
<div class="row">
<div class="col-sm-4">
<div class="card border-secondary">
<div class="card-header"><h5>Events<span class="badge badge-
secondary">${number_events}</span></h5></div>
<div class="card-body">
<p class="card-text">Events we organize for our customers.</p>
<br>
<a href="events" class="btn btn-outline-secondary">View Events</a>
</div>
</div>
</div>
<div class="col-sm-4">
<div class="card border-secondary">
<div class="card-header"><h5>Services<span class="badge badge-
warning">${number_services}</span></h5></div>
<div class="card-body">
```

Surana College

Dept of Computer Sci.

Project- VI Sem BCA - 2020

Page | 37

```
<p class="card-text">Services offered by us to our customers.</p>
<br>
<a href="services" class="btn btn-outline-warning">View Services</a>
</div>
</div>
</div>
<div class="col-sm-4">
<div class="card border-secondary">
<div class="card-header"><h5>Users<span class="badge badge-
success">${number_users}</span></h5></div>
<div class="card-body">
<p class="card-text">Customers registered with us using this portal.</p>
<a href="users" class="btn btn-outline-success">View Customers</a>
</div>
</div>
</div>
</div>
<br>
<div class="row">
<div class="col-sm-4">
<div class="card border-secondary">
<div class="card-header"><h5>Uploads<span class="badge badge-
info">${number_uploads}</span></h5></div>
<div class="card-body">
<p class="card-text">These are profiles uploaded by people willing to join the
team.</p>
<a href="upLoads" class="btn btn-outline-info">View Uploads</a>
</div>
</div>
</div>
<!-- <div class="col-sm-4">
<div class="card border-secondary">
<div class="card-header"><h5>Team<span class="badge badge-
danger">${number_team}</span></h5></div>
<div class="card-body">
<p class="card-text">Our team members.</p>
<br>
<a href="#" class="btn btn-outline-danger">View Team</a>
</div>
</div>
</div> --%>
<div class="col-sm-4">
<div class="card border-secondary">
<div class="card-header"><h5>Gallery<span class="badge badge-
primary">${number_gallery}</span></h5></div>
<div class="card-body">
<p class="card-text">View Gallery.</p>
<br>
<a href="gallery" class="btn btn-outline-primary">Manage Gallery</a>
</div>
</div>
</div>
</div>
<br>
</div>
```

Surana College

Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
<br><br><br><br>
```

```
<%@ include file="common/footer.jspf" %>
```

Page | 38 Change password:

```
<%@ include file="common/header.jspf" %>
<%@ include file="common/navigation.jspf" %>
<br>
<div class="container">
<h3>Change Password</h3>
<br>
<form:form method="post" commandName="changepassword">

<div class="row">
<div class="col">
<label for="password">Current Password</label>
<input type="password" id="password" name="password" class="form-control"
placeholder="Enter Current Password" required>
</div>
<div class="col">

</div>
</div>
<br>
<div class="row">
<div class="col">
<label for="newPassword">New Password</label>
<input type="password" id="newPassword" name="newPassword" class="form-control"
placeholder="Enter New Password" required>
</div>
<div class="col">

</div>
</div>
<br>

<div class="form-group">
<button type="submit" class="btn btn-success" id="update">Change Password</button>
</div>
</form:form>
<br>
<c:if test="${not empty message and messagetype == 'error'}">
<div style="color: red">
<h5>${message}</h5>
</div>
</c:if>

<c:if test="${not empty message and messagetype == 'success'}">
<div style="color: green">
<h5>${message}</h5>
</div>
</c:if>
```

Surana College

Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
</c:if>
</div>
```

```
<%@ include file="common/footer.jspf" %>
```

Messages:

Page | 39

```
<%@ include file="common/header.jspf" %>
<%@ include file="common/navigation.jspf" %>
<br>
<div class="container">
<h3>Messages</h3>
</div>
<br>
<div class="container">
<c:if test="${not empty messages}">
<table class="table table-striped">
<thead>
<tr>
<th>Message #</th>
<th>Email</th>
<th>Mobile</th>
<th>Message</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<c:forEach items="${messages}" var="message">
<tr>
<td style="width: 150px !important;">${message.id}</td>
<td style="width: 100px !important;">${message.email}</td>
<td style="width: 100px !important;">${message.mobile}</td>
<td style="width: 500px !important;">${message.message}</td>
<td><a type="button" class="btn btn-link" href="#">Acknowledge</a></td>
<td><a type="button" class="btn btn-link" href="#">Delete</a></td>
</tr>
</c:forEach>
</tbody>
</table>
</c:if>

<c:if test="${empty messages}">
<div style="color: green">
<h5>There are no files messages.</h5>
</div>
</c:if>

<br>
<a href = "dashboard" class="btn btn-link">Back to Dashboard</a>
</div>

<br><br>

<%@ include file="common/footer.jspf" %>
```

Surana College

Dept of Computer Sci.

Project- VI Sem BCA - 2020

Uploads:

Page | 40

```
<%@ include file="common/header.jspf" %>
<%@ include file="common/navigation.jspf" %>
<br>
<div class="container">
<h3>Uploads</h3>
</div>
<br>
<div class="container">
<c:if test="${not empty uploads}">
<table class="table table-striped">
<thead>
<tr>
<th>Upload #</th>
<th>File Name</th>
<th></th>
</tr>
</thead>
<tbody>
<c:forEach items="${uploads}" var="upload">
<tr>
<td>${upload.id}</td>
<td>${upload.fileName}</td>
<td><a type="button" class="btn btn-link"
href="download?fileName=${upload.fileName}">Download</a></td>
</tr>
</c:forEach>
</tbody>
</table>
</c:if>

<c:if test="${empty uploads}">
<div style="color: green">
<h5>There are no files uploaded.</h5>
</div>
</c:if>

<br>
<a href="dashboard" class="btn btn-link">Back to Dashboard</a>
</div>

<br><br>

<%@ include file="common/footer.jspf" %>
```

WEB CONTROLLERS:

ChangePasswordController:

```
package com.elegant.springboot.web.controller;
```


Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
import javax.servlet.http.HttpServletRequest;
```

```
import javax.servlet.http.HttpSession;
```

```
import javax.validation.Valid;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.ui.ModelMap;
```

```
import org.springframework.validation.BindingResult;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
```

```
import org.springframework.web.bind.annotation.RequestMethod;
```

```
import org.springframework.web.bind.annotation.RequestParam;
```

```
import com.elegant.springboot.web.model.ChangePassword;
```

```
import com.elegant.springboot.web.model.User;
```

```
import com.elegant.springboot.web.service.UserService;
```

```
@Controller
```

```
public class ChangePasswordController {
```

```
@Autowired
```

```
UserService userService;
```

```
@RequestMapping(value = "/changepassword", method =  
RequestMethod.GET)
```

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
public String showChangePassword(ModelMap model, HttpServletRequest  
request) {
```

```
try {
```

```
Page | 42 HttpSession session = request.getSession();
```

```
if(session != null) {
```

```
if(session.getAttribute("name") != null) {
```

```
String userName = session.getAttribute("name").toString();
```

```
model.put("name", userName);
```

```
ChangePassword changePassword = new ChangePassword();
```

```
model.addAttribute("changepassword", changePassword);
```

```
return "changepassword";
```

```
}
```

```
}
```

```
} catch (Exception e) {
```

```
System.err.println("Error: " + e.getMessage());
```

```
e.printStackTrace();
```

```
throw e;
```

```
}
```

```
return "login";
```

```
}
```

```
@RequestMapping(value = "/changepassword", method =  
RequestMethod.POST)
```

```
public String changePassword (ModelMap model, @RequestParam String  
password, @RequestParam String newPassword, @Valid ChangePassword  
changePassword, HttpServletRequest request, BindingResult result) {
```

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
try {
    HttpSession session = request.getSession();
    if(session != null) {
        if(session.getAttribute("name") != null) {
            Integer id = Integer.parseInt(session.getAttribute("userid").toString());
            User updateUser = userService.getUser(id);
            if(updateUser.getPassword().equalsIgnoreCase(newPassword)) {
                model.addAttribute("message", "Current password and new password should
                not be same.");
                model.addAttribute("messagetype", "error");
                return "changepassword";
            }
            updateUser.setPassword(newPassword);
            updateUser = userService.updateUser(id, updateUser);
            if(updateUser != null && updateUser.getId() != null) {
                model.addAttribute("message", "Your password is changed successfully!");
                model.addAttribute("messagetype", "success");
                session.setAttribute("name", null);
                session.invalidate();
                return "login";
            } else {
                model.addAttribute("message", "Somthing went wrong. We couldn't change
                your password.");
                model.addAttribute("messagetype", "error");
            }
        }
    }
}
```

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
}  
  
}  
  
} catch (NumberFormatException e) {  
System.err.println("Error: " + e.getMessage());  
e.printStackTrace();  
throw e;  
}  
return "changepassword";  
}  
  
}
```

Page | 44

Email Sender:

```
package com.elegant.springboot.web.controller;
```

```
import java.util.ArrayList;
```

```
import java.util.List;
```

```
import java.util.Properties;
```

```
import javax.mail.Message;
```

```
import javax.mail.MessagingException;
```

```
import javax.mail.PasswordAuthentication;
```

```
import javax.mail.Session;
```

```
import javax.mail.Transport;
```

```
import javax.mail.internet.InternetAddress;
```

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
import javax.mail.internet.MimeMessage;
```

```
import com.elegant.springboot.web.model.Order;
```

Page | 45

```
public class EmailSender {
```

```
    public static Properties props = new Properties();
```

```
    static {
```

```
        String smtp = "smtp.gmail.com";
```

```
        String port = "587";
```

```
        String senderEmail = "manager.elegantevents@gmail.com";
```

```
        String senderPassword = "!Elegantevents1";
```

```
        props.put("mail.smtp.auth", "true");
```

```
        props.put("mail.smtp.starttls.enable", "true");
```

```
        props.put("mail.smtp.host", smtp);
```

```
        props.put("mail.smtp.port", port);
```

```
        props.put("sender.email", senderEmail);
```

```
        props.put("sender.password", senderPassword);
```

```
    }
```

```
    public static boolean sendResetPasswordEmailMessage(String to, String  
newpassword) throws MessagingException {
```

```
        String subject = "Elegant Events - Password Reset";
```

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
String message = String.format("<h4>Your New  
Password</h4><p><b>%s</b></p>", newpassword);  
return sendEmail(to, subject, message);
```

Page | 46 }

```
public static boolean sendOrderDetailsEmailMessage(List<String> tos, Order  
order) throws MessagingException {
```

```
String subject = "Elegant Events - Order Details";
```

```
String message = String.format("<h4>Your Order Details : %s </h4>" +
```

```
"<h5>Order # : %s</h5> " +
```

```
"<h5>Status : %s</h5>" +
```

```
"<h5>Cost : ₹ %s</h5>" +
```

```
"<h5>Address : %s</h5>" +
```

```
"<h5>Email : %s</h5>" +
```

```
"<h5>Mobile : %s</h5>" +
```

```
"<h5>Date : %s</h5>" +
```

```
"<hr>" + "<a href=\"http://localhost:9091/\">Visit our website</a>",
```

```
order.getName(), order.getId(), order.getStatus(), order.getCost(),  
order.getAddress(), order.getEmail(), order.getMobile(),  
order.getOrderDate());
```

```
return sendEmail(tos, subject, message);
```

```
}
```

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
public static boolean sendEmail(String to, String subject, String message)
throws MessagingException {
```

```
    System.out.println("Sending email to " + to);
```

```
    Session session = createSession(props);
```

```
    //create message using session
```

```
    MimeMessage mimeTypeMessage = new MimeMessage(session);
```

```
    List<String> tos = new ArrayList<String>();
```

```
    tos.add(to);
```

```
    prepareEmailMessage(mimeTypeMessage, tos, subject, message, props);
```

```
    //sending message
```

```
    Transport.send(mimeTypeMessage);
```

```
    System.out.println("Done");
```

```
    return true;
```

```
}
```

```
public static boolean sendEmail(List<String> tos, String subject, String message)
throws MessagingException {
```

```
    System.out.println("Sending email to " + tos);
```

```
    Session session = createSession(props);
```

```
    //create message using session
```

```
    MimeMessage mimeTypeMessage = new MimeMessage(session);
```

```
    prepareEmailMessage(mimeTypeMessage, tos, subject, message, props);
```

```
    //sending message
```

```
    Transport.send(mimeTypeMessage);
```

```
    System.out.println("Done");
```

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
return true;
```

```
}
```

Page | 48

```
private static void prepareEmailMessage(MimeMessage message, List<String>  
tos, String title, String html, Properties emailConfig) throws  
MessagingException {
```

```
message.setContent(html, "text/html; charset=utf-8");
```

```
message.setFrom(new  
InternetAddress(emailConfig.getProperty("sender.email")));
```

```
for (String to: tos) {
```

```
message.setRecipients(Message.RecipientType.TO, InternetAddress.parse(to));
```

```
}
```

```
message.setSubject(title);
```

```
}
```

```
private static Session createSession(Properties emailConfig) {
```

```
Session session = Session.getInstance(emailConfig, new  
javax.mail.Authenticator() {
```

```
protected PasswordAuthentication getPasswordAuthentication() {
```

```
return new PasswordAuthentication(emailConfig.getProperty("sender.email"),  
emailConfig.getProperty("sender.password"));
```

```
}
```

```
});
```

```
return session;
```

```
}
```


Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

}

Page | 49

Event Controller:

```
package com.elegant.springboot.web.controller;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.ui.ModelMap;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
```

```
import org.springframework.web.bind.annotation.RequestMethod;
```

```
import com.elegant.springboot.web.model.Event;
```

```
import com.elegant.springboot.web.service.EventService;
```

```
@Controller
```

```
public class EventController {
```

```
@Autowired
```

```
EventService eventService;
```

```
@RequestMapping(value = "/list-events", method = RequestMethod.GET)
```

```
public String showEvents(ModelMap model, HttpServletRequest request) {
```

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
try {  
    model.put("events", eventService.getAllEvents());  
    model.put("name", EventUtils.getLoggedInUserName(request.getSession()));  
} catch (Exception e) {  
    System.err.println("Error: " + e.getMessage());  
    e.printStackTrace();  
    throw e;  
}  
return "list-events";  
}
```

Page | 50

```
@RequestMapping(value = "/add-event", method = RequestMethod.GET)  
public String showAddEventPage(ModelMap model, HttpServletRequest  
request) {  
    try {  
        Event event = new Event("TEST", "Test Event", "This is Test Event Description",  
1000.0);  
        eventService.addEvent(event);  
        model.addAttribute("event", event);  
        model.put("name", EventUtils.getLoggedInUserName(request.getSession()));  
    } catch (Exception e) {  
        System.err.println("Error: " + e.getMessage());  
        e.printStackTrace();  
        throw e;  
    }  
}
```

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
return "list-events";  
}
```

Page | 51

```
@RequestMapping(value = "/eventdetails", method = RequestMethod.GET)  
public String showEventDetails(ModelMap model, HttpServletRequest request)  
{  
    try {  
        model.put("name", EventUtils.getLoggedInUserName(request.getSession()));  
    } catch (Exception e) {  
        System.err.println("Error: " + e.getMessage());  
        e.printStackTrace();  
        throw e;  
    }  
    return "eventdetails";  
}  
  
}
```

File Upload Exception Handler:

```
package com.elegant.springboot.web.controller;  
  
import  
org.apache.tomcat.util.http.fileupload.FileUploadBase.FileSizeLimitExceededEx  
ception;  
  
import org.springframework.web.bind.annotation.ControllerAdvice;  
import org.springframework.web.bind.annotation.ExceptionHandler;
```

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
import
org.springframework.web.multipart.MaxUploadSizeExceededException;

import org.springframework.web.multipart.MultipartException;

import org.springframework.web.servlet.ModelAndView;

import org.springframework.web.servlet.mvc.support.RedirectAttributes;
```

Page | 52

```
@ControllerAdvice

public class FileUploadExceptionHandler {

    @ExceptionHandler(MultipartException.class)

    public ModelAndView handleError1(MultipartException e, RedirectAttributes
    redirectAttributes) {

        ModelAndView mv = new ModelAndView();

        mv.setViewName("error");

        mv.addObject("message", e.getCause().getCause().getMessage());

        mv.addObject("messagetype", "error");

        return mv;

    }
```

```
@ExceptionHandler(MaxUploadSizeExceededException.class)

public ModelAndView handleError2(MaxUploadSizeExceededException e,
    RedirectAttributes redirectAttributes) {

    ModelAndView mv = new ModelAndView();

    mv.setViewName("error");

    mv.addObject("message", e.getCause().getLocalizedMessage());

    mv.addObject("messagetype", "error");
```

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
return mv;  
}
```

Page | 53

```
@ExceptionHandler(FileSizeLimitExceededException.class)  
  
public ModelAndView handleError3(FileSizeLimitExceededException e,  
RedirectAttributes redirectAttributes) {  
  
    ModelAndView mv = new ModelAndView();  
  
    mv.setViewName("error");  
  
    mv.addObject("message", e.getCause().getLocalizedMessage());  
  
    mv.addObject("messagetype", "error");  
  
    return mv;  
  
}
```

Forgot Password Controller:

```
package com.elegant.springboot.web.controller;  
  
  
  
import javax.mail.MessagingException;  
import javax.servlet.http.HttpServletRequest;  
  
  
import org.apache.commons.lang3.StringUtils;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.ModelMap;  
import org.springframework.web.bind.annotation.RequestMapping;
```

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.RequestParam;
```

Page | 54

```
import com.elegant.springboot.web.model.User;
```

```
import com.elegant.springboot.web.service.UserService;
```

```
@Controller
```

```
public class ForgotPasswordController {
```

```
@Autowired
```

```
UserService userService;
```

```
@RequestMapping(value = "/forgotpassword", method =  
RequestMethod.GET)
```

```
public String showForgotPassword(ModelMap model, HttpServletRequest  
request) {
```

```
return "forgotpassword";
```

```
}
```

```
@RequestMapping(value = "/resetpassword", method =  
RequestMethod.POST)
```

```
public String performResetPassword(ModelMap model, @RequestParam  
String username, HttpServletRequest request) throws Exception {
```

```
try {
```

```
String message = "";
```

```
if(StringUtils.isEmpty(username)) {
```

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
User user = null;

user = userService.getUserByEmail(username);

if(user == null) {
    user = userService.getUserByMobile(username);
}

if(user != null) {
    //User found. Generate password.
    String newPassword = "";
    for(char ch : EventUtils.generatePassword(6)) {
        newPassword += ch;
    }

    System.out.println("New Password\n"+newPassword);
    user.setPasword(newPassword);

    //Reset password
    userService.updateUser(user.getId(), user);

    //Send email
    boolean emailSent = false;

    emailSent = EmailSender.sendResetPasswordEmailMessage(user.getEmail(),
        newPassword);

    if(emailSent) {
        message = "Your password is reset and sent to email or mobile.";
        model.addAttribute("message", message);
        model.addAttribute("messagetype", "success");
    } else {
```

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
message = "Something went wrong. We couldn't reset password. Make sure  
your email is valid.";
```

```
model.addAttribute("message", message);
```

```
Page | 56 model.addAttribute("messagetype", "error");
```

```
}
```

```
} else {
```

```
message = "We didn't find registered user with Email or Mobile you entered.  
Sign up if you are new user.";
```

```
model.addAttribute("message", message);
```

```
model.addAttribute("messagetype", "error");
```

```
}
```

```
}
```

```
} catch (MessagingException e) {
```

```
System.err.println("Error: " + e.getMessage());
```

```
e.printStackTrace();
```

```
throw e;
```

```
}
```

```
return "forgotpassword";
```

```
}
```

```
}
```

Login Controller:

```
package com.elegant.springboot.web.controller;
```


Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
import javax.servlet.http.HttpServletRequest;
```

```
import javax.servlet.http.HttpSession;
```

Page | 57

```
import org.apache.commons.lang3.StringUtils;
```

```
import org.apache.tomcat.util.codec.binary.Base64;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.ui.Model;
```

```
import org.springframework.ui.ModelMap;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
```

```
import org.springframework.web.bind.annotation.RequestMethod;
```

```
import org.springframework.web.bind.annotation.RequestParam;
```

```
import com.elegant.springboot.web.model.User;
```

```
import com.elegant.springboot.web.service.StorageService;
```

```
import com.elegant.springboot.web.service.UserService;
```

```
@Controller
```

```
public class LoginController {
```

```
    @RequestMapping(value = "/signin", method = RequestMethod.GET)
```

```
    public String login(Model model) {
```

```
        return "/login";
```

```
    }
```

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

@Autowired

UserService userService;

Page | 58

@Autowired

StorageService storageService;

@RequestMapping(value = "/login", method = RequestMethod.POST)

public String performLogin(@RequestParam(value = "username", required = false) String loginUserNameInput,

@RequestParam(value = "password", required = false) String password, ModelMap model, HttpServletRequest request) throws Exception {

try {

HttpSession session = request.getSession();

String errorMessage = "";

User user = userService.getUserByInput(loginUserNameInput);

if(user != null) {

if(((user.getUserName() != null &&

loginUserNameInput.equalsIgnoreCase(user.getUserName()))

|| (user.getEmail() != null &&

loginUserNameInput.equalsIgnoreCase(user.getEmail()))

|| (user.getMobile() != null &&

loginUserNameInput.equalsIgnoreCase(user.getMobile()))

&& (user.getPasword() != null && password.equals(user.getPasword())) {

String userName = StringUtils.upperCase(user.getFirstName());

if(session != null && null == session.getAttribute("name")) {

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
session = request.getSession(true);
session.setAttribute("name", userName);
session.setAttribute("role", user.getRole());
session.setAttribute("userid", user.getId());
if(user.getPicture() != null) {
String encodedfile = new String(Base64.encodeBase64(user.getPicture()));
session.setAttribute("profilepic", encodedfile);
} else {
session.setAttribute("profilepic", null);
}
}
if(user.getPicture() == null || user.getPicture().length == 0) {
session.setAttribute("profilepic", null);
}
return "welcome";
} else {
errorMessage = "Incorrect password. Try again or reset password using Forgot
password.";
}
} else {
errorMessage = "We didn't find registered user with Email or Mobile you
entered. Sign up if you are new user.";
}
model.addAttribute("message", errorMessage);
model.addAttribute("messagetype", "error");
```

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
return "login";
} catch (Exception e) {
System.err.println("Error: " + e.getMessage());
e.printStackTrace();
throw e;
}
}
}
```

Page | 60

Order Controller:

```
package com.elegant.springboot.web.controller;

import java.sql.Date;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.List;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;
import javax.validation.Valid;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.RequestMapping;
```

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
import org.springframework.web.bind.annotation.RequestMethod;
```

```
import org.springframework.web.bind.annotation.RequestParam;
```

Page | 61

```
import com.elegant.springboot.web.model.Event;
```

```
import com.elegant.springboot.web.model.Order;
```

```
import com.elegant.springboot.web.model.Payment;
```

```
import com.elegant.springboot.web.model.PlaceOrder;
```

```
import com.elegant.springboot.web.model.Service;
```

```
import com.elegant.springboot.web.service.EventService;
```

```
import com.elegant.springboot.web.service.OrderService;
```

```
import com.elegant.springboot.web.service.PaymentService;
```

```
import com.elegant.springboot.web.service.ServiceService;
```

```
@Controller
```

```
public class OrderController {
```

```
@Autowired
```

```
OrderService orderService;
```

```
@Autowired
```

```
EventService eventService;
```

```
@Autowired
```

```
ServiceService serviceService;
```

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

@Autowired

PaymentService paymentService;

Page | 62

```
@RequestMapping(value = "/orderdetails", method = RequestMethod.GET)
public String showOrderDetails(@RequestParam Integer id, ModelMap model)
{
    try {
        Order order = orderService.getOrder(id);
        model.put("order", order);
    } catch (Exception e) {
        System.err.println("Error: " + e.getMessage());
        e.printStackTrace();
        throw e;
    }
    return "order";
}
```

```
@RequestMapping(value = "/placeorder", method = RequestMethod.GET)
public String placeOrder(@RequestParam String code, @RequestParam String
type, ModelMap model, HttpServletRequest request) {
    try {
        HttpSession session = request.getSession();
        if(session != null) {
            if(session.getAttribute("name") != null) {
                String userName = session.getAttribute("name").toString();
            }
        }
    }
}
```

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
model.put("name", userName);

String orderName = null;

String cost = null;

if(type != null && !type.isEmpty()) {
    if(type.equalsIgnoreCase("event")) {
        Event event = eventService.getEventByCode(code);
        orderName = event.getName() + " - " + event.getDescription();
        cost = String.valueOf(event.getCost());
    } else if(type.equalsIgnoreCase("service")) {
        Service service = serviceService.getServiceByCode(code);
        orderName = service.getName() + " - " + service.getDescription();
        cost = String.valueOf(service.getCost());
    }
}

PlaceOrder order = new PlaceOrder();

order.setCode(code);

if(cost != null && !cost.isEmpty()) {
    order.setCost(cost);
}

model.addAttribute("ordername", orderName);
model.addAttribute("code", code);
model.addAttribute("cost", cost);
model.addAttribute("placeorder", order);

return "placeorder";
}
```

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
}  
} catch (Exception e) {  
    System.err.println("Error: " + e.getMessage());  
    e.printStackTrace();  
    throw e;  
}  
return "login";  
}
```

Page | 64

```
@RequestMapping(value = "/placeorder", method = RequestMethod.POST)  
public String performPlaceOrder(ModelMap model, HttpServletRequest  
request, @Valid PlaceOrder placeOrder, BindingResult result) throws Exception  
{  
    try {  
        if (result.hasErrors()) {  
            return "placeorder";  
        }  
        HttpSession session = request.getSession();  
        if (session != null && session.getAttribute("userid") != null) {  
            Integer userId = Integer.parseInt(session.getAttribute("userid").toString());  
            double cost = Double.parseDouble(placeOrder.getCost());  
            Payment payment = new Payment();  
            payment.setUserId(userId);  
            payment.setType("CreditCard");  
            payment.setNameOnTheCard(placeOrder.getNameOnCard());
```


Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
payment.setExpYear(placeOrder.getExpYear());
payment.setExpMonth(placeOrder.getExpMonth());
payment.setCvv(placeOrder.getCvv());
payment.setCardNumber(placeOrder.getCardNumber());
payment.setBrand("Visa");
payment.setAmount(cost);
Payment savedPayment = paymentService.addPayment(payment);
if(savedPayment != null && savedPayment.getId() != null) {
String eventServiceName = null;
if(placeOrder.getType() != null && !placeOrder.getType().isEmpty()) {
if(placeOrder.getType().equalsIgnoreCase("event")) {
Event event = eventService.getEventByCode(placeOrder.getCode());
eventServiceName = event.getName() + " - " + event.getDescription();
} else if(placeOrder.getType().equalsIgnoreCase("service")) {
Service service = serviceService.getServiceByCode(placeOrder.getCode());
eventServiceName = service.getName() + " - " + service.getDescription();
}
}
Order newOrder = new Order();
newOrder.setUserId(userId);
newOrder.setType(placeOrder.getType());
newOrder.setStatus("Confirmed");
newOrder.setPaymentId(savedPayment.getId());
newOrder.setMobile(placeOrder.getMobile());
newOrder.setEmail(placeOrder.getEmail());
```

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

Page | 66

```
newOrder.setCost(cost);
newOrder.setCode(placeOrder.getCode());
newOrder.setAddress(placeOrder.getAddress());
newOrder.setContact(placeOrder.getFirstName() + " "+
placeOrder.getLastName());
newOrder.setName(eventServiceName);
String orderDate = placeOrder.getOrderDate();
SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd");
java.util.Date dt = format.parse(orderDate);
newOrder.setOrderDate(new Date(dt.getTime()));
newOrder.setCreatedOn(new Date(new java.util.Date().getTime()));
Order savedOrder = orderService.addOrder(newOrder);
boolean emailSent = false;
List<String> toEmails = new ArrayList<String>();
toEmails.add(savedOrder.getEmail());
emailSent = EmailSender.sendOrderDetailsEmailMessage(toEmails,
savedOrder);

if(!emailSent && savedOrder.getId() != null) {
String message = "Your order # "+ savedOrder.getId()+" is placed successfully.
But we couldnt send email to : "+ savedOrder.getEmail();
model.addAttribute("message", message);
model.addAttribute("messagetype", "error");
} else if(emailSent && savedOrder.getId() == null) {
String message = "Your order # "+ savedOrder.getId()+" is placed successfully.
We have sent an email with order details to : "+ savedOrder.getEmail();
```

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
model.addAttribute("message", message);
model.addAttribute("messagetype", "success");
}
return "redirect:orderdetails?id="+savedOrder.getId();
}
}
} catch (Exception e) {
System.err.println("Error: " + e.getMessage());
e.printStackTrace();
throw e;
}
return "placeorder";
}
```

Page | 67

```
@RequestMapping(value = "/myorders", method = RequestMethod.GET)
public String showMyOrders(ModelMap model, HttpServletRequest request) {
try {
HttpSession session = request.getSession();
if(session != null) {
if(session.getAttribute("name") != null) {
String userName = session.getAttribute("name").toString();
Integer userId = Integer.parseInt(session.getAttribute("userid").toString());
model.put("name", userName);
model.put("myorders", orderService.getOrdersByUserId(userId));
return "myorders";
}
```

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
}  
  
}  
  
} catch (NumberFormatException e) {  
    System.err.println("Error: " + e.getMessage());  
    e.printStackTrace();  
    throw e;  
}  
return "login";  
}  
  
}
```

Services Controller:

```
package com.elegant.springboot.web.controller;  
  
import javax.servlet.http.HttpServletRequest;  
  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.ModelMap;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
  
@Controller  
public class ServicesController {  
  
    @RequestMapping(value = "/list-services", method = RequestMethod.GET)
```

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
public String showServices(ModelMap model, HttpServletRequest request) {  
    try {  
        model.put("name", EventUtils.getLoggedInUserName(request.getSession()));  
    } catch (Exception e) {  
        System.err.println("Error: " + e.getMessage());  
        e.printStackTrace();  
        throw e;  
    }  
    return "list-services";  
}
```

Page | 69

```
@RequestMapping(value = "/servicedetails", method = RequestMethod.GET)  
public String showServiceDetails(ModelMap model, HttpServletRequest  
request) {  
    try {  
        model.put("name", EventUtils.getLoggedInUserName(request.getSession()));  
    } catch (Exception e) {  
        System.err.println("Error: " + e.getMessage());  
        e.printStackTrace();  
        throw e;  
    }  
    return "servicedetails";  
}  
}
```

User Controller:

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
package com.elegant.springboot.web.controller;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import javax.validation.Valid;
```

```
import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.stereotype.Controller;
```

```
import org.springframework.ui.ModelMap;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
```

```
import org.springframework.web.bind.annotation.RequestMethod;
```

```
import org.springframework.web.bind.annotation.RequestParam;
```

```
import com.elegant.springboot.web.model.SignUpUser;
```

```
import com.elegant.springboot.web.model.User;
```

```
import com.elegant.springboot.web.service.UserService;
```

```
@Controller
```

```
public class UserController {
```

```
@Autowired
```

```
UserService userService;
```

```
@RequestMapping(value = "/signup", method = RequestMethod.GET)
```

```
public String showSignup(ModelMap model, HttpServletRequest request) {
```

```
try {
```

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
SignUpUser signUpUser = new SignUpUser();
model.addAttribute("signupuser", signUpUser);
} catch (Exception e) {
    System.err.println("Error: " + e.getMessage());
    e.printStackTrace();
    throw e;
}
return "signup";
}
```

Page | 71

```
@RequestMapping(value = "/signup", method = RequestMethod.POST)
public String performSignup(ModelMap model, @RequestParam String
password, @RequestParam String confirmPassword, @Valid SignUpUser
signUpUser, HttpServletRequest request) {
    try {
        String message = null;
        User existingUser = null;
        existingUser = userService.getUserByEmail(signUpUser.getEmail());
        if(existingUser == null) {
            existingUser = userService.getUserByMobile(signUpUser.getMobile());
        }
        if(existingUser != null) {
            message = "Email or Mobile is already registered. Try different one.";
            model.addAttribute("message", message);
            model.addAttribute("messagetype", "error");
        }
    }
}
```

Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
} else if(existingUser == null) {  
    User user = new User();  
    user.setFirstName(signUpUser.getFirstName());  
    user.setLastName(signUpUser.getLastName());  
    user.setEmail(signUpUser.getEmail());  
    user.setMobile(signUpUser.getMobile());  
    user.setAddress(signUpUser.getAddress());  
    user.setPassword(password);  
    user.setUserName(signUpUser.getFirstName());  
    user.setRole("Customer");  
    User savedUser = userService.addUser(user);  
    if(savedUser != null && savedUser.getId() != null) {  
        message = "Registration succesful!";  
        model.addAttribute("message", message);  
        model.addAttribute("messagetype", "success");  
    } else {  
        message = "Registration failed!. Something went wrong.";  
        model.addAttribute("message", message);  
        model.addAttribute("messagetype", "error");  
    }  
}  
  
} catch (Exception e) {  
    System.err.println("Error: " + e.getMessage());  
    e.printStackTrace();  
    throw e;
```


Surana College
Dept of Computer Sci.

Project- VI Sem BCA - 2020

```
}  
return "signup";  
}  
}
```

TESTING

Software testing is a critical element of software quality assurance and represents the ultimate review of specifications, design, coding. The engineer creates a series of test cases that are intended to “demolish” the software that has been built. In fact, testing is the one step in this software engineering process that could be viewed as destructive rather than constructive. The testing process is divided into three major components as follows

- Unit testing
- Integration testing
- Validation testing

Unit Testing

The software is tested using the unit test method. Unit testing focuses verification effort on the smallest unit of software design module. Using the procedural design description as a guide, important control parts are tested to uncover errors within the boundary of the module. Every module in the project is checked for verification.

Integration Testing

This testing is the systematic technique for constructing the program structure by performing the test in each module and later combining the entire individual module to very large program.

Validation Testing

Validation testing is the process of testing the input. Whether the given inputs are valid or invalid. As part of validation testing, we may use statistical testing to test the program performance and reliability and to check how it works under operational conditions.

CONCLUSION

There are all formal and cultural events taking place everyday in and around us. Event management companies play a major role in this. Doing so, consumption of time and man power on manual works like maintaining records of orders are redundant. This web application creates a user-friendly interface to reduce such redundant tasks. The web application can not only be used for an Event management company but with minor changes, it can be converted into any organization to manage their records.

FUTURE ENHANCEMENTS

1. Encrypt user password in database
2. Enable user to connect with Google / Facebook account
3. File upload – increased data
4. Payments - use real-time payment gateways
5. QR code scanner for website
6. Implementation of search feature across website
7. Collect feedback/review of service from customers

BIBLIOGRAPHY

- <https://www.tutorialspoint.com>
- www.youtube.com/edureka
- www.youtube.com/freecodecamp/springboot