

CHAPTER 1

1.1. Introduction

Events create opportunities for people to connect with an area, spend time together, celebrate and experience the diversity of cultures and foster creativity and innovation. They allow a community to come alive and provide an opportunity for a destination to showcase its tourism experience and increase economic activity. Events contribute significantly to community building, lifestyle and leisure enhancement, cultural development, volunteer participation, fundraising and economic development. Most importantly, events create a sense of fun and vibrancy, resulting in a strong sense of community connectivity, pride and a sense of place.

1.2. Scope of the capstone project

Problem Statement:

As per my survey in current market there is no online system to take the registrations of any events so event organizers' waste the time to take the registrations. In current market organizers can add the event in web portal.

Objectives:

Reduce the time of user.

Reduce the manual work.

Easy to use.

Access from any location.

Easily event registration.

No wastage of money.

Capstone project description:

The purpose of this project is to create “Event Mitra”. In this project the publisher wants to publish their events and user will take the registration for the event competition. Publisher needs to register and create events. After creating the events any user can see events list and participate. Admin can get the publisher registration list and created events list. To implement this project, we will use full stack languages like PHP, HTML, CSS, Bootstrap, JavaScript, MySQL.

Capstone project Deliverables:

- Admin Module

Admin login: - In this using username and password admin can login.

View publisher registration: - In this admin can view all the publisher’s registration list.

Manage and view events: - In this admin can view event list, and they can approve and disapprove events.

- Publisher modules

Publisher registration: - In this publisher register under admin.

Publisher login: -In this publisher can login after the registration.

Create event and competition: - In this publisher can create event with proper details.

View applied participants list: - In this publisher can view the registered participants list for their event.

- User modules

View events list: - In this user can view the events list.

Apply for events: - In this user can apply for the events.

Key milestones: -

- Admin modules

Admin login

View publisher registration

Manage and view events

- Publisher modules

Publisher Registration

Publisher Login

Create event and Competition

View applied participants list

- User modules

View events list

Apply for events

Constraints: - Timelines for project is limited.

To acquire knowledge and skilled labour.

Cost is limited.

Require limited employees for project.

Estimated Capstone project Duration: - 12 weeks (72days)

Estimated Capstone project cost: -

The total estimated cost of this project 46,000

Chapter 2

2.1 Capstone project planning

2.1.1 Work Break-Down Structure

- Admin modules

Admin login: - In this using username and password admin can login.

View publisher registration: - In this admin can view the publisher registration list.

Manage and view events: - In this admin can view event list, and they can approve and disapprove events.

- Publisher modules

Publisher registration: - In these publishers register under admin.

Publisher login: -In these publishers can login after the registration.

Create event and competition: - In this publisher can create event with proper details.

View applied participants list: -In this publisher can view the registered participants list for their event.

- User modules

View event list: - In this user can view the events list.

Apply for events: - In this user can apply for the events.

Activities & Tasks: -

- Admin login

Task 1: -Admin must be login with their correct details.

Task 2: -Admin login form must validate properly with entered details.

- View publisher registration

Task 1: -Admin must view the publisher registration for appropriate details.

Task 2: -Admin must validate the view publisher registration page properly with correct entered details.

- Manage and view events

Task 1: -Admin must view event list.

Task 2: -Admin can approve and disapprove events.

- Publisher registration
 - Task 1: -Publisher must register with proper details.
 - Task 2: -Publisher should provide proper details according to the validation.
- Publisher login
 - Task 1: -Publisher must be login with their correct details.
 - Task 2: -Publisher form must validate properly with entered details.
- Create event and competition
 - Task 1: -Publishers must create and provide required details of particular events.
 - Task 2: -Publisher must provide category to the event accordingly.
- View applied participants list
 - Task 1: -Publisher must view list of applied participants for various events.
 - Task 2: -Publisher must rectify the details of participants.
- View event list
 - Task 1: -User must view the events list to participate.
- Apply for events
 - Task 1: -User must enter the correct details for participation.
 - Task 2: -User must rectify the terms and conditions of particular event.

2.1.2 Timeline Development – Schedule

Activities & Task

- Admin login (Within 6 days 42 hours)
 - Task 1: -Admin must be login with their correct details.
 - Task 2: -Admin login form must validate properly with entered details.
 - Collect requirement gathering and requirement analysis to create admin login form.
 - To design admin login form using HTML, CSS, Bootstrap.
 - Validation of admin login form using JavaScript.
 - Perform database connection using PHP and MySQL.
 - Testing admin login form to check validation and database.
 - This whole task done by Divya Patil.
- Publisher Registration (Within 6 days 42 hours)
 - Task 1: -Publisher must register with their proper details.
 - Task 2: -Publisher should provide proper details according to the validation.

Collect requirement gathering and requirement analysis to create publisher registration form.

To design publisher registration form using HTML, CSS, Bootstrap.

Validation of publisher registration form using JavaScript.

Perform database connection using PHP and MySQL.

Testing publisher registration form to check validation and database.

This whole task is done by Jyotigouda Patil.

- Publisher Login (Within 5 days 35 hours)

Task 1: -Publisher must be login with their correct details.

Task 2: -Publisher form must validate properly with entered details.

Collect requirement gathering and requirement analysis to create publisher login form.

To design publisher login form using HTML, CSS, Bootstrap.

Validation of publisher login form using JavaScript.

Perform database connection using PHP and MySQL.

Testing publisher login form to check validation and database.

This whole task is done by Akkavva Dange.

- View publisher registration (Within 4 days 28 hours)

Task 1: -Admin must view the publisher registration list for appropriate details.

Task 2: -Admin must validate the view publisher registration page properly with correct entered details.

Collect requirement gathering and requirement analysis to create view publisher registration list.

To design view publisher registration list using HTML, CSS, Bootstrap.

Validation of view publisher registration list using JavaScript.

Perform database connection using PHP and MySQL.

Testing view publisher registration list to check validation and database.

This whole task is done by Shridhar Kumbhar.

- Create event and Competition (Within 12 days 84 hours)

Task 1: -Publishers must create and provide required details of particular events.

Task 2: -Publisher must provide category to the event accordingly.

Collect requirement gathering and requirement analysis to create event and competition page.

To design create event and competition page using HTML, CSS, Bootstrap.

Validation of create event and competition page using JavaScript.

Perform database connection using PHP and MySQL.

Testing creates event and competition page to check validation and database.

This whole task is done by Divya Patil.

- Manage and view events (Within 10 days 70 hours)

Task 1: -Admin must view event list.

Task :2 -Admin can approve and disapprove events.

Collect requirement gathering and requirement analysis to create manage and view events list.

To design manage and view events list using HTML, CSS, Bootstrap.

Validation of manage and view events list using JavaScript.

Perform database connection using PHP and MySQL.

Testing manages and view events list to check validation and database.

This whole task is done by Akkavva Dange.

- User view events list (Within 5 days 35 hours)

Task 1: -User must view the events list for participation.

Collect requirement gathering and requirement analysis to create user view events list page.

To design view events list using HTML, CSS, Bootstrap.

Validation of view events list using JavaScript.

Perform database connection using PHP and MySQL.

Testing view events list to check validation and database.

This whole task is done by Shridhar Kumbhar.

- Apply for events (Within 8 days 56 hours)

Task 1: -User must enter the correct details for participation.

Task 2: -User must rectify the terms and conditions for particular event.

Collect requirement gathering and requirement analysis to create apply for events page.

To design apply for events page using HTML, CSS, Bootstrap.

Validation of apply for events page using JavaScript.

Perform database connection using PHP and MySQL.

Testing apply for events page to check validation and database.

This whole task is done by Jyotigouda Patil.

- View applied participants list (Within 4 days 28 hours)
 - Task 1: -Publisher must view list of applied participants for various events.
 - Task 2: -Publisher must rectify the details of participants.
- Collect requirement gathering and requirement analysis to create view applied participants list page.
- To design view applied participants list using HTML, CSS, Bootstrap.
- Validation of view applied participants list using JavaScript.
- Perform database connection using PHP and MySQL.
- Testing view applied participants list to check validation and database.
- This whole task is done by Akkavva Dange.

2.1.3 Cost Break-down Structure

Analyse your Work Breakdown Structure

- Admin modules
 - Admin login
 - View publisher registration
 - Manage and view events
- Publisher modules
 - Publisher Registration
 - Publisher Login
 - Create event and Competition
 - View applied participants list
- User modules
 - View events list
 - Apply for events

Estimate the labour cost of work:

| Modules | Task | Time | | | Total cost (hour*amount per hour) |
|------------------------------|--|----------------------|-------|--------------------|---|
| | | Hours Per Task | Hours | Amount Per hour | |
| Admin login | Admin must be login with their correct details. | 18 | 42 | ₹60 | 42*60= ₹2,520 |
| | Admin login form must validate properly with entered details. | 24 | | | |
| Publisher registration | Publisher must register with their proper details. | 17 | 42 | ₹60 | 42*60= ₹2,520 |
| | Publisher should provide proper details according to the validation. | 25 | | | |
| Publisher login | Publisher must login with their correct details. | 15 | 35 | ₹60 | 35*60= ₹2,100 |
| | Publisher form must validate properly with entered details. | 20 | | | |
| View publisher registration | Admin must view the publisher registration list for appropriate details. | 12 | 28 | ₹60 | 28*60= ₹1,680 |
| | Admin must validate this page properly. | 16 | | | |
| Create event and competition | Publisher must create and provide required details of particular events. | 44 | 84 | ₹60 | 84*60= ₹5,040 |
| | Publisher must provide category to the events. | 40 | | | |
| Manage and view events | Admin must view event list. | 30 | 70 | ₹60 | 70*60= ₹4,200 |
| | Admin can approve & disapprove events. | 40 | | | |
| User view events list | User must view events list for participation. | 35 | 35 | ₹60 | 35*60= ₹2,100 |
| | User must enter correct details | 35 | 56 | ₹60 | 56*60= ₹3,360 |

| | | | | | |
|---------------------------|--|----|----|-----|--------------|
| Apply for Events | User must rectify terms & conditions for particular event. | 21 | | | |
| View applied participants | Publisher must view the participants list. | 12 | 28 | ₹60 | 28*60=₹1,680 |
| | Publisher must rectify details of participants | 16 | | | |

Table.:2.1.3.1

Estimate the cost of materials:

The cloud server cost is of ₹5000.

PHP Designer license key of rupees ₹3450.

XAMPP Severe is of free Source.

Overhead Cost:

If we need extra cost for live server.

Build Contingency into your CBS:

As per the estimated cost we need deliver the project.

Final-Check:

| Title | Cost (Rupees) |
|------------------|----------------------|
| Estimated Cost | ₹46,000 |
| Labour Cost | ₹25,200 |
| Cost of material | ₹8,800 |
| Net Profit | ₹12,000 |

Table.:2.1.3.2

2.1.4 Capstone project Risks assessment

Home page must display Admin login, Register, Login, Post event, view events, Contact US, About Us and Home all of these must be displayed and work properly.
Admin login form must work without any error.

Admin must be able to see publisher registered list with all the details, status to activate or inactivate and delete option.

Admin must be able to see all the events list with all the details, status to activate or inactivate and delete option.

Publisher registration form must work and be registered without any error.

Publisher login form must work without any error.

Publisher post event form work efficiently without error in any field.

Publisher posted events list must display all the events list with all the details, status.

Registration list of the participants must be visible to the publishers.

User must be able to view all the Events with details and picture.

User must be able to book for the events in which they are interested.

2.2 REQUIREMENT SPECIFICATION

2.2.1 Functional requirements:

- Admin Aspects

Admin login into the system

View the publishers list

View the posted events

Managing the events list

Admin can logout

- Publisher Aspects

Registration

Login

Create events and competition

View applied participants list

Logout

- User Aspects

View events list

Book for events

2.2.2 Non-functional requirements:

- Admin

Id

Name

Mobile

Email

Username

Password

Status

- Events

Id

Event_name

Event_date

Event_location

Event-image

Event_details

User id

Status

- Event_booking

Id

Event_id

Name

Phone

Email

- User

Id

Name

Phone

Email

Username

Password

Status

2.2.3 User input

- Admin login

Username

Password

- Publisher registration

Your name

Your phone

Your email

Username

Password

- Publisher login

Username

Password

- Create events and competition

Event name

Event date

Event location

Event image

Event details

- Apply For events

Your name

Your phone

Your email

2.2.4 Technical Constraints:

Requirements for developing: -

Average I3 processor.

Average 4GB RAM.

Average 512GB hard disk.

Hardware requirement for deployment: -

Average I3 processor.

Average 2GB RAM.

Average 256GB storage.

Software requirement for development: -

OS (Window's, MAC, Linux).

XAMP Server.

VS Code.

Chrome.

Software requirement for deployment: -

OS.

Chrome.

2.3 DESIGN SPECIFICATION

2.3.1 Chosen System Design:

System Architecture:

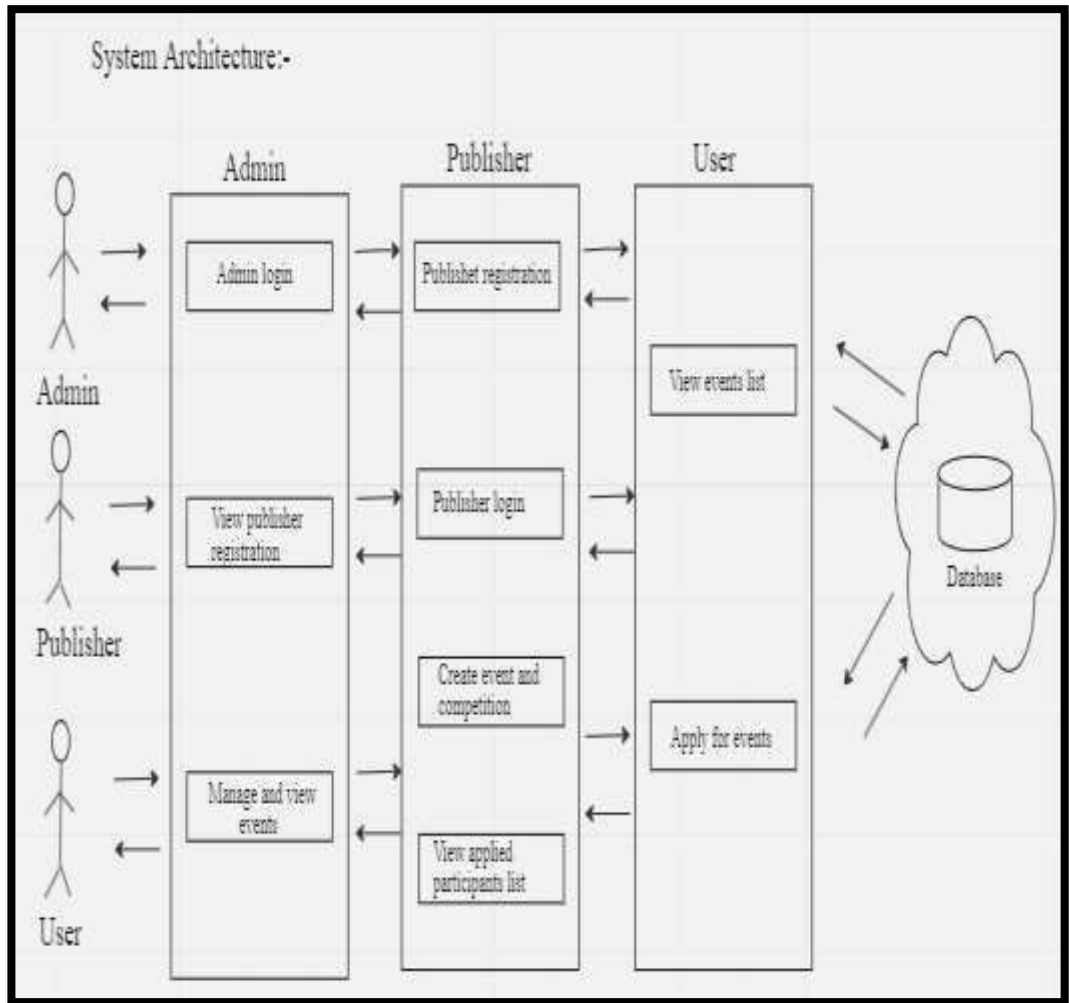


Fig.:2.3.1.1

System architecture diagrams provide a visual illustration of a system's various components and show how they communicate and interact with each other. This allows for a clear understanding of how the system works and how it can be improved. Identification of requirements that have an impact on the application's structure is the architecture's main objective. We have 3 main modules, are as follows Admin module, Publisher module, User module.

2.3.2 Discussion of Alternative design

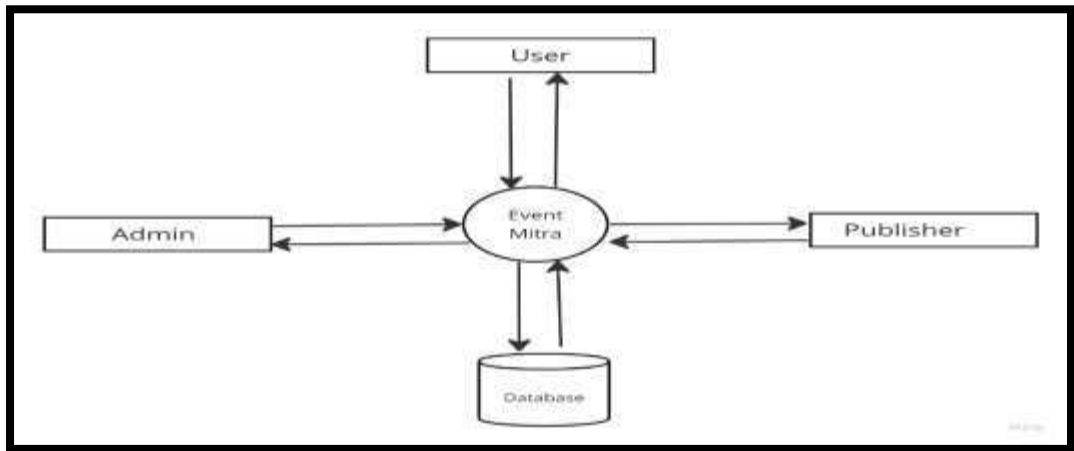


Fig.:2.3.2.1

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination.

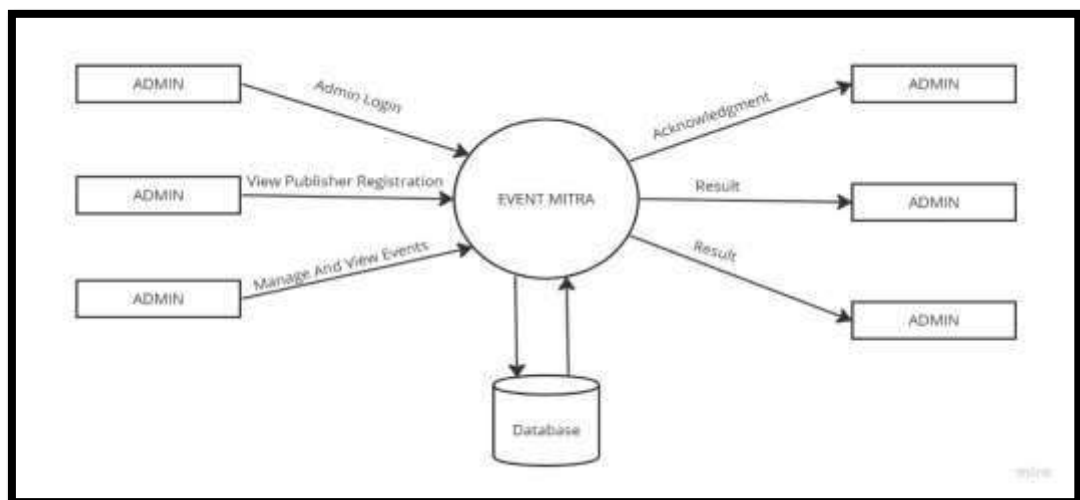


Fig.:2.3.2.2 Admin

A data flow diagram (DFD) is a graphical or visual depiction that describes how data is moved through an organization's activities using a standardised set of symbols and notations. The path that data follows between external entities, processes, and data repositories is known as data flow. Arrows are used to depict the interaction between the other components, and they are usually labelled with a brief data name. Such as “Manage and view events”.

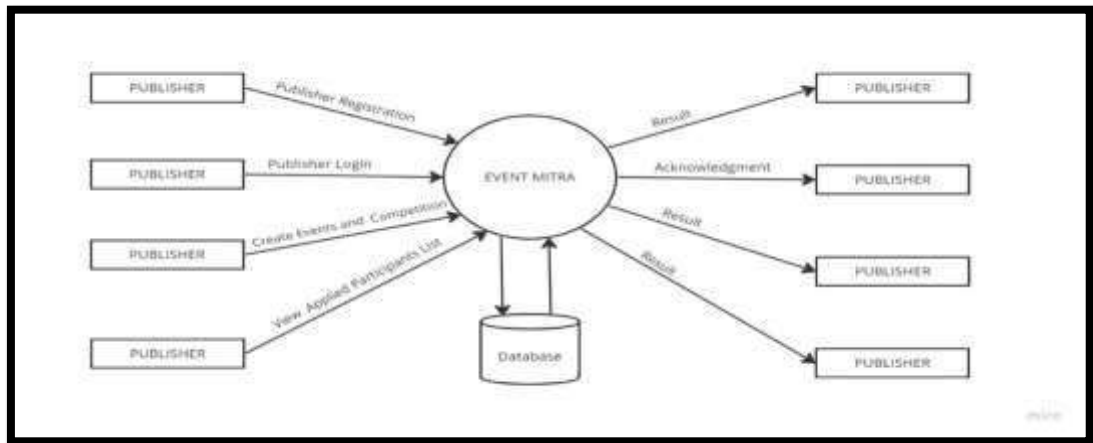


Fig.:2.3.2.3 Publisher

A data flow diagram (DFD) is a graphical or visual depiction of a business's processes using a standardised set of symbols and notations. The path that data follows between external entities, processes, and data repositories is known as data flow. It depicts the interaction with the other components and is represented by arrows, which are usually labelled with a brief data name, such as "create events and competition"

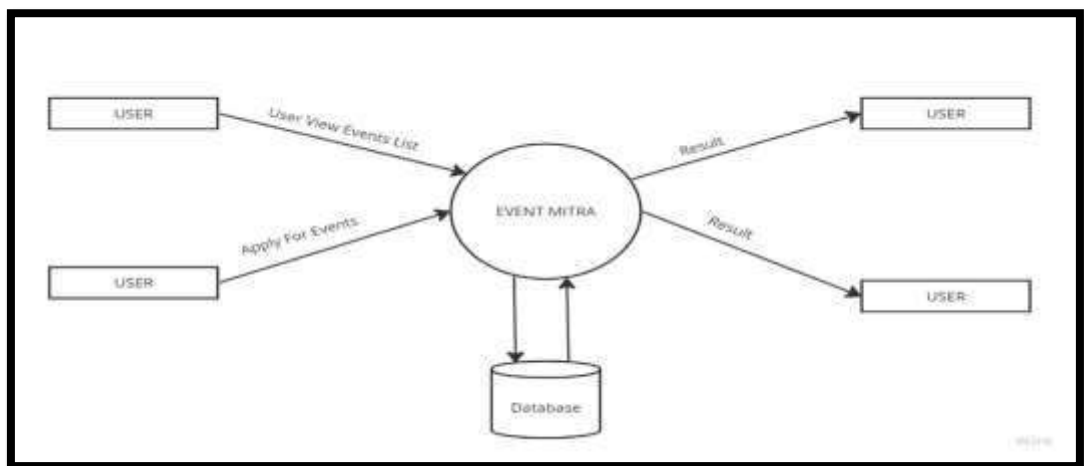


Fig.:2.3.2.4 User

A data flow diagram (DFD) is a graphical or visual depiction of a business's activities through data movement that uses a standardised set of symbols and notations. Data flow is the path that information travels as it moves between external entities, processes, and data repositories. It represents the interface with the other elements and is represented by arrows, which are often labelled with a brief data name, such as "User view events list".

2.3.3 Detailed Description of Components/subsystems

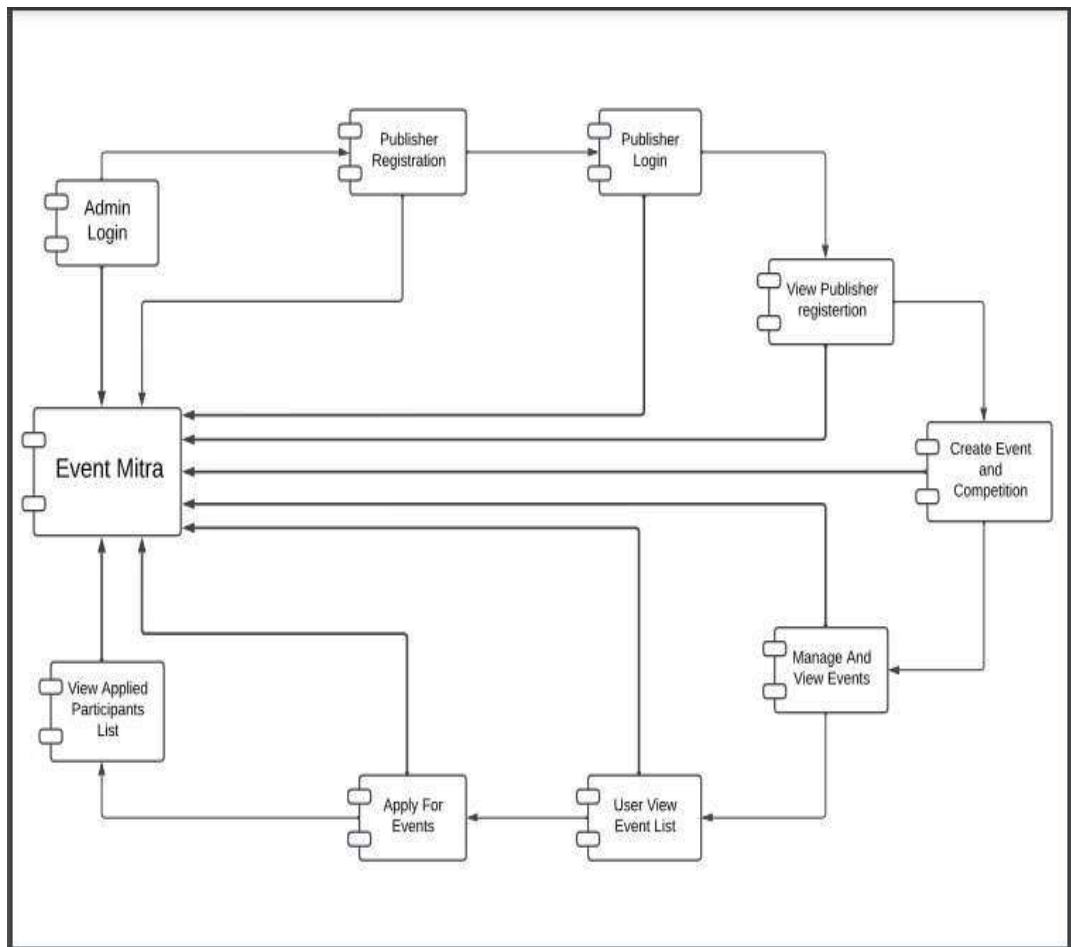


Fig.:2.3.3.1

A component diagram is used to illustrate the connections between various system components. Component diagrams display the organisation of the software system, which includes information on the interfaces and dependencies of the programme components. Here are the elements. Here, we have displayed the module dependencies in this diagram. Admin login manages the view publisher registration list and manage and view events, publisher registration is followed up by publisher login and create event and competition,

2.3.4 Component 1-n

- Admin login: -

In admin login using username and password admin can login. In this admin can manage all the sub modules like he can see all the publishers who posted the events & he can able to approve the events. Admin can manage and view the events.

- Publisher registration: -

In this publisher can register under the admin. In registration we need fill some fields after filling the fields we can register.

- Publisher login: -

In this using particular username and password publisher can login. After login publisher can create the events.

- View publisher registration: -

In this admin can view the publisher registration list. In these lot of publishers are present who created the events for competition.

- Create events and competition: -

In this, after login of publisher he can need to create the events. In this publisher should create events for competition and should provide the required details of particular events.

- Manage and view events: -

In this admin can view event list, and they can approve and disapprove events.

In this when admin can make events status active then only user can view the events, otherwise events can't visible to the users. So, admin can manage and approve the events.

- User view events list: -

In this user can view the created events for their participation. In this user can view all the created events.

- Apply for events: -

In this user can apply for the events. In this user need to fill the correct details for applying the events.

- View applied participants list: -

In this publisher can view the registered participants list for their event.

CHAPTER- 3

3.1 APPROACH AND METHODOLOGY

3.1.1 Discuss the Technology:

Web technology

A place connected to the internet, where a company, organization, etc. Puts information that can be found on the World Wide Web.

Types of web technology: -

Browsers

HTML and CSS

Programming Languages

Frameworks

Web Servers

Databases

Protocols

Lastly, data Formats

Advantages

We can access from any location.

No data loss.

Data can be recovered.

Be available every time.

We can save time.

Cloud based technology

To live the project, we need to purchase cloud sever.

Cloud based technology is the use of software and services via the internet. These applications commonly include data storage, networking, servers and databases. User can access their cloud hosted tools with any device that is connected to the internet.

Types of cloud-based technology

Private cloud: -

There is no interference from third parties in a private cloud.

Public cloud: -

Public clouds are managed by third-party vendors that offer on-demand cloud computing services to multiple organizations.

Hybrid cloud: -

Hybrid clouds allow the integration of private and public cloud features into a single IT infrastructure.

Multi cloud: -

Multi clouds bring together multiple public clouds from different cloud service providers.

Advantages

Usability and accessibility

Security

Cost efficient

Convenient sharing of files

Automation

Open-Source web technology

For our project we not require to purchase any software's and libraries.

XAMPP

XAMPP is a cross-platform and open-source tool, which makes it an ideal choice of web developers. It is the acronym of X-cross platform, Apache, MySQL, PHP, and Perl.

PHP

PHP is an open-source scripting language used for creating dynamic and interactive web pages and various digital platforms.

PhpMyAdmin

PhpMyAdmin is an open source and free administration tool for MySQL.

Advantages

Community-Driven Reliability

Community-Driven Security

Low Cost on an Ongoing Basis

Better, Community-Based Collaboration

3.1.2 Methodologies

Agile methodology is a process for managing a project that involves constant collaboration and working in iterations. Agile project management works off the basis that a project can be continuously improved upon throughout its life cycle, with changes being made quickly and responsively. Agile is one of the most popular approaches to project Management due to it 's flexible, adaptability to change, and high level of customer input.

Phases of Agile model: -

Requirement Gathering

Design the requirements

Construction or Iterations

Testing and Quality assurance

Deployment

Feedback

Agile Diagram:-



Fig.:3.1.2.1

3.1.3 Use Cases

Use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behaviour (what), and not the exact method of making it happen.

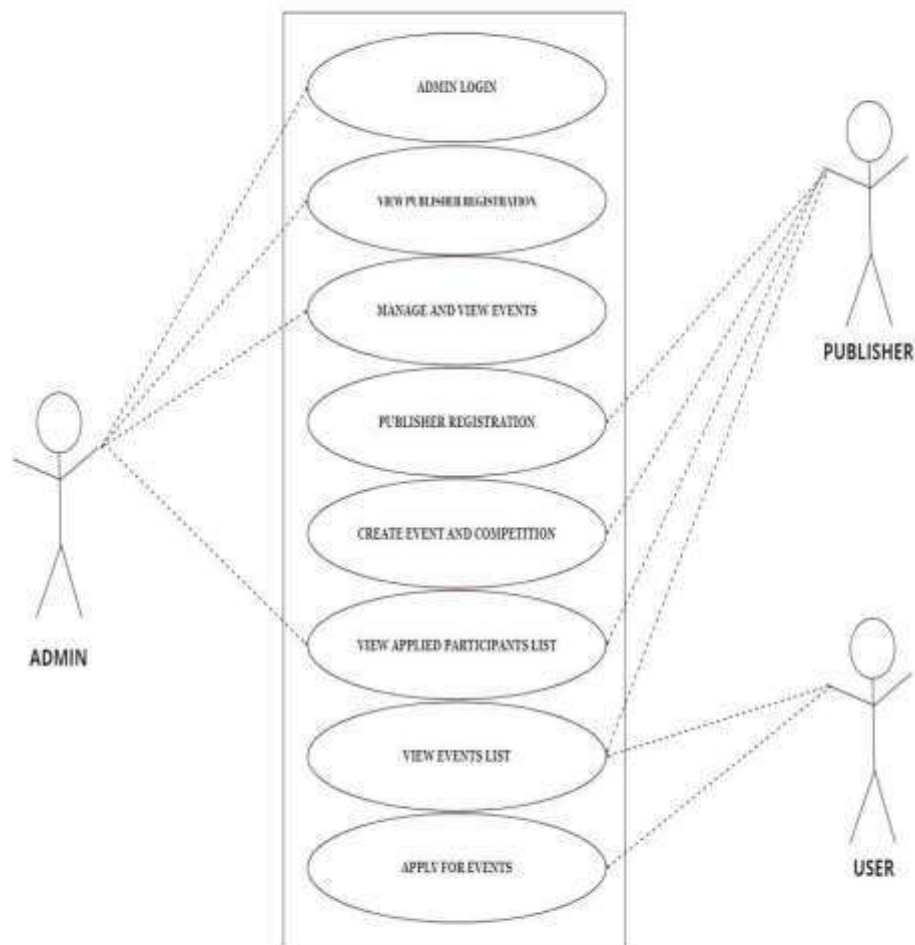


Fig.:3.1.3.1

3.1.4 Programming

- **HTML**

It stands for hypertext markup language. It is client-side scripting language. It is used to create static web pages. File extension of html is .html & html contains predefined tags.

Advantages

It is easy to learn.

Every browser supports HTML Language.

HTML is light weighted and fast to load.

HTML has many tags and attributes which can short your line of code.

- **CSS**

CSS stands for Cascading Style Sheets. CSS is the language we use to style an html document. CSS describes how html elements should be displayed. It can control the layout of multiple web pages all at once. External stylesheets are stored in CSS files.

Advantages

Helps in making creative web pages by making them simple to use.

Improve the browsing speed.

It can be used on various devices.

Wider variety of design options.

- **PHP**

PHP stands for Hypertext Pre-processor. PHP is a server-side scripting language designed specifically for web development. It is open-source which means it is free to download and use. It is very simple to learn and use. The extension for php is '.php'.

Advantages

It's open-source and free from cost.

It is platform-independent.

It has less learning curve because it is simple and straightforward to use. It helps in managing code easily.

▪ **Bootstrap**

Bootstrap is a free and open-source framework for web development. It uses HTML, CSS, and JavaScript to create responsive and mobile-friendly websites and web applications. It provides a collection of syntax for template designs that make web development easier and faster.

Advantages

Open source

Easy to use

Save lots of time

Compatible with browser

▪ **JavaScript**

JavaScript is the most popular lightweight, interpreted compiled programming language. It can be used for both Client-side as well as Server-side developments. JavaScript also known as a scripting language for web pages. It is use for validation purpose.

Advantages

Fast speed

Easy to learn

Versatility

Popularity

3.1.5 Analysis:

- **Existing system:** - As per my survey in current market there is no online system to take the registrations of any events so event organizers' waste the time to take the registrations. In current market organizers can add the event in web portal.
- **Proposed System:** - We will implement your project name. The purpose of this project is to create "EVENT MITRA". In this project publisher wants to publish their events and user will take the registration for the event competition. Publisher needs to register and create events. After creating the events any user can see events list and participate. Admin can get the publisher registration list and created events. To implement this project, we will use full stack languages like PHP, HTML, CSS, Bootstrap, JavaScript, MySQL.

3.1.6 Process Design:

Tables

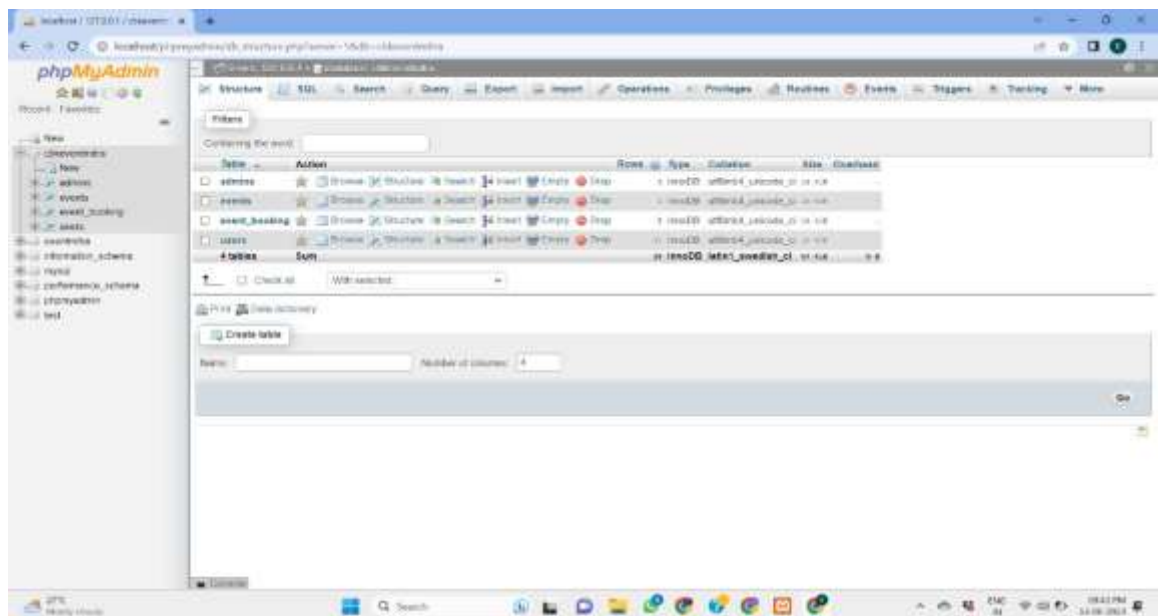


Fig.:3.1.6.1

Admin

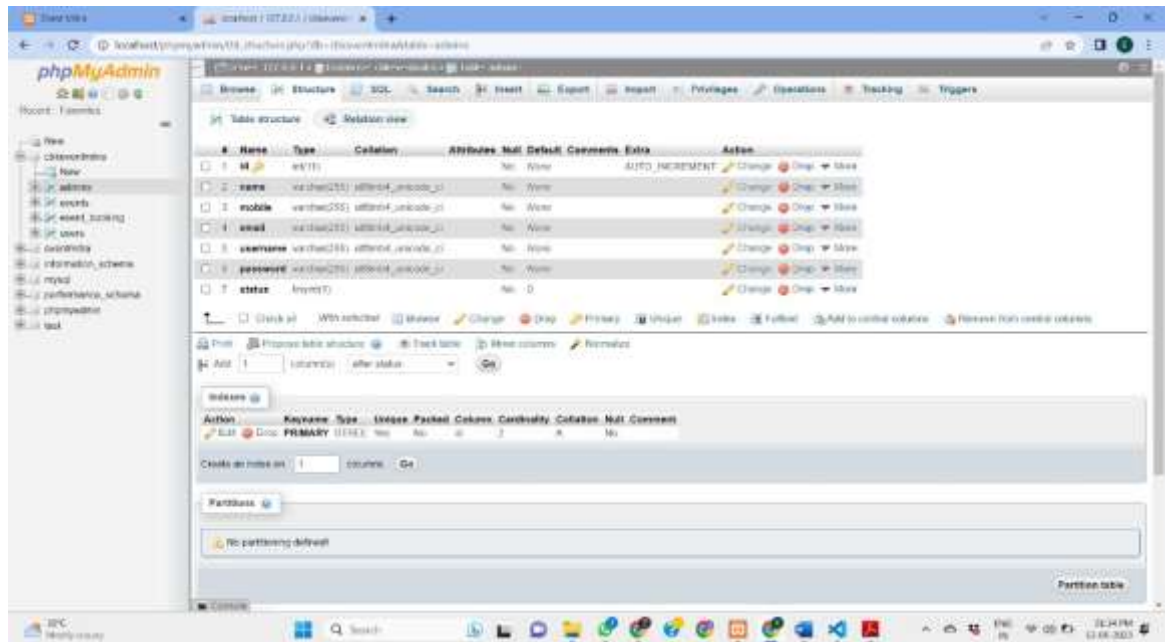


Fig.:3.1.6.2

Events

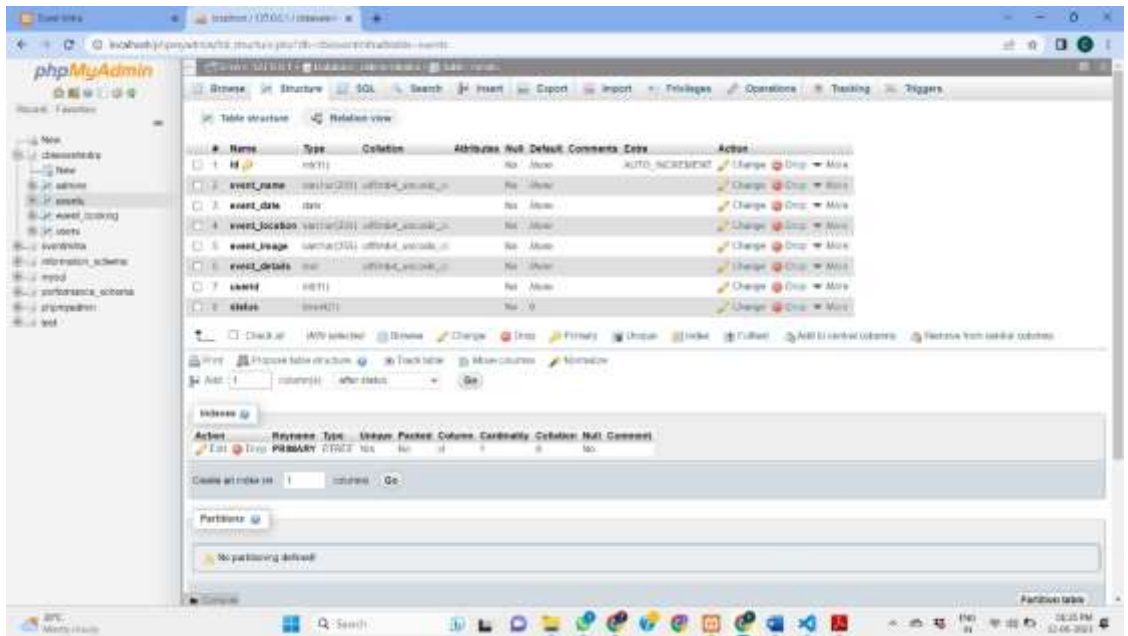


Fig.:3.1.6.3

Event_booking

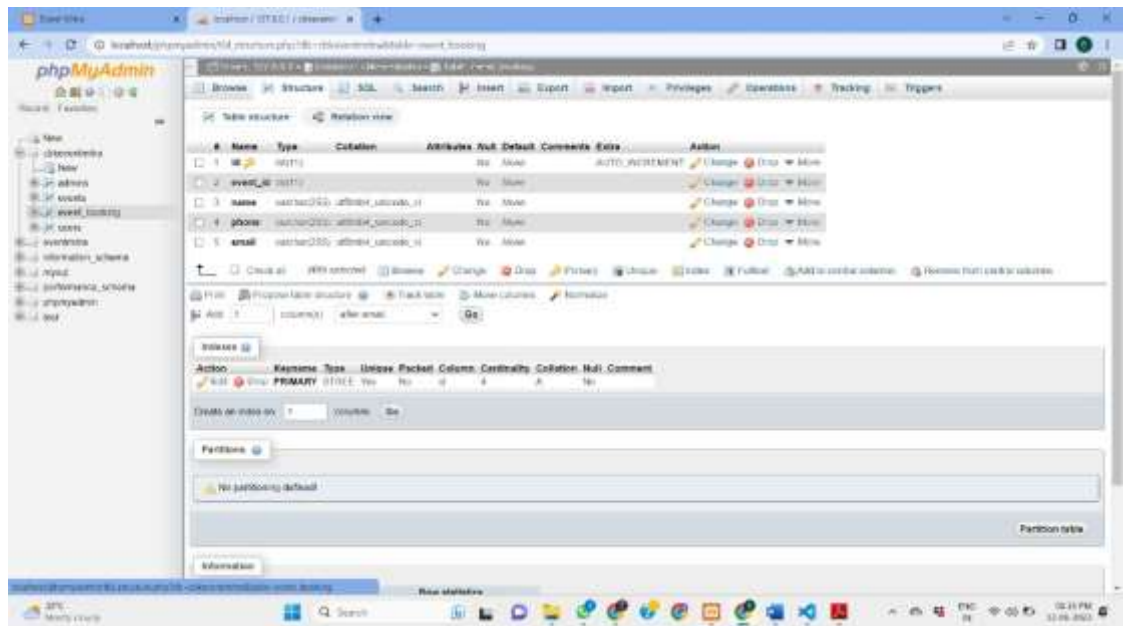


Fig.3.1.6.4

User

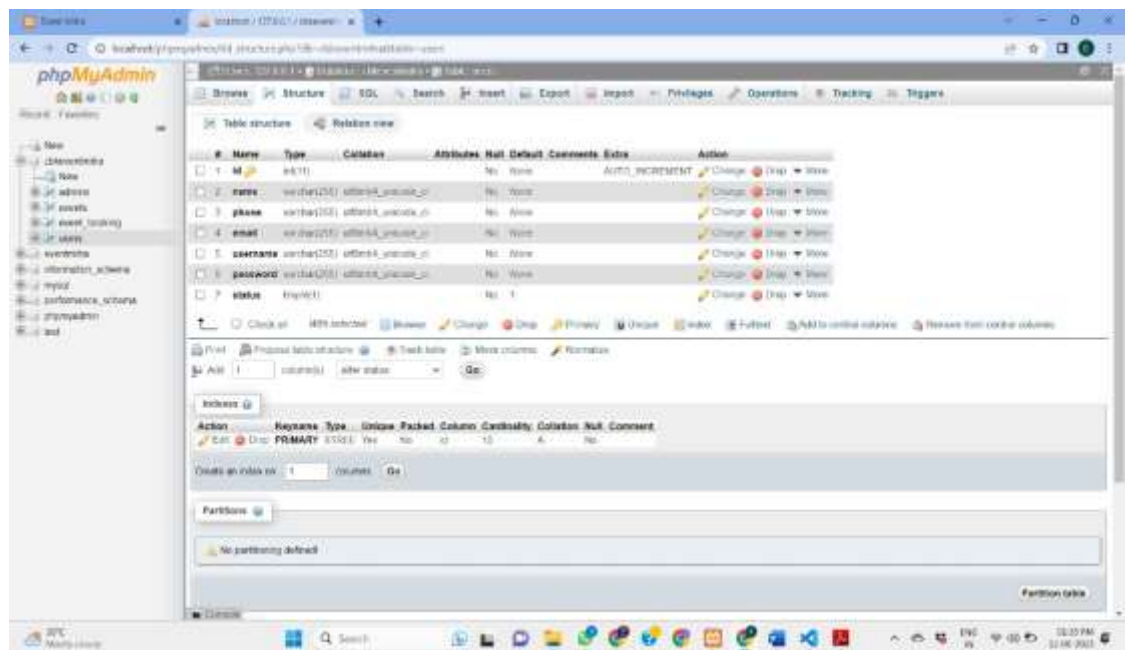


Fig.3.1.6.5

3.1.7 Product Design:

Home page

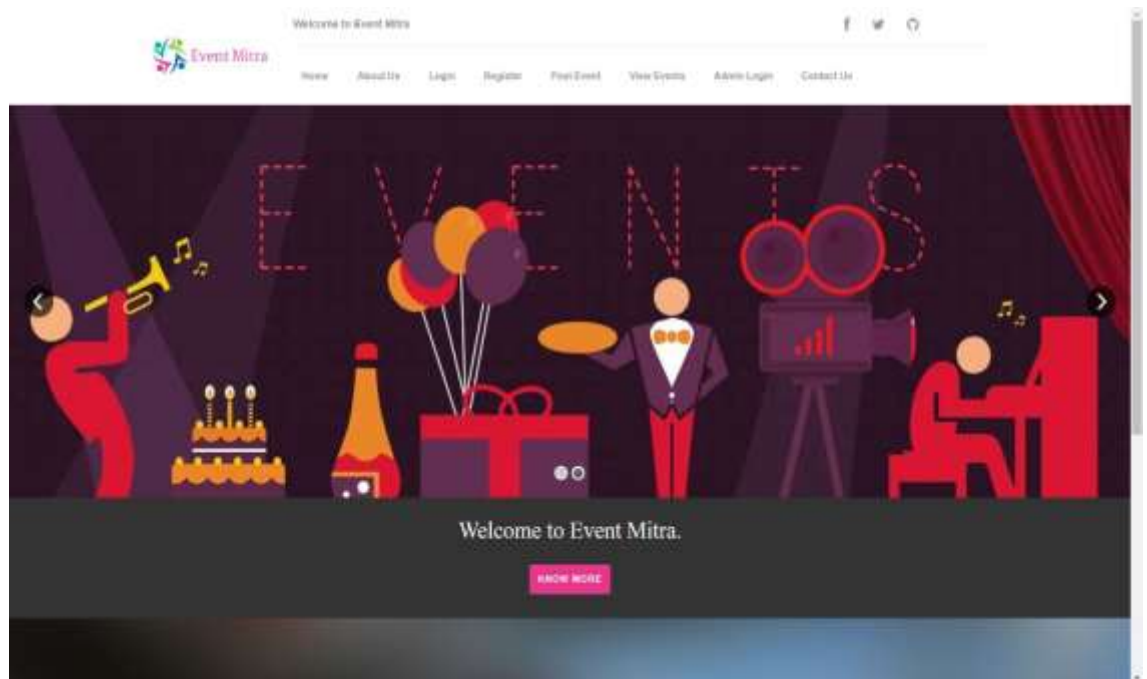


Fig.:3.1.7.1

Admin Login

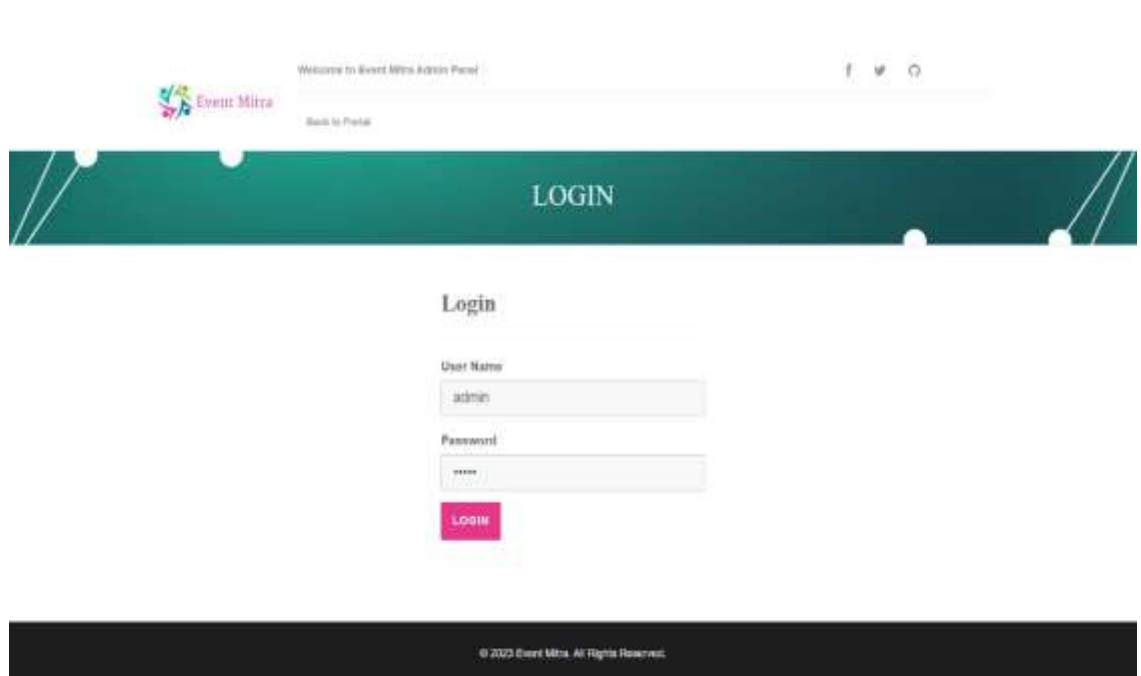


Fig.:3.1.7.2

Publisher Registration

The screenshot displays the Event Mitra website's registration page. At the top, a navigation bar includes the site logo, a welcome message, social media icons, and links for Home, About Us, Login, Register, Post Event, View Events, Admin Login, and Contact Us. Below this is a large green banner with the word "REGISTER" in white. The main content area is titled "Register" and contains a form with the following fields: "Your Name" (filled with "Divya"), "Your Phone" (filled with "9617382460"), "Your Email" (filled with "divya@gmail.com"), "User Name" (filled with "divya"), and "Password" (filled with "****"). A pink "REGISTER NOW!" button is positioned below the form, followed by a link that says "Have account? Login Here". The footer is a black bar with the text "© 2020 Event Mitra. All Rights Reserved."

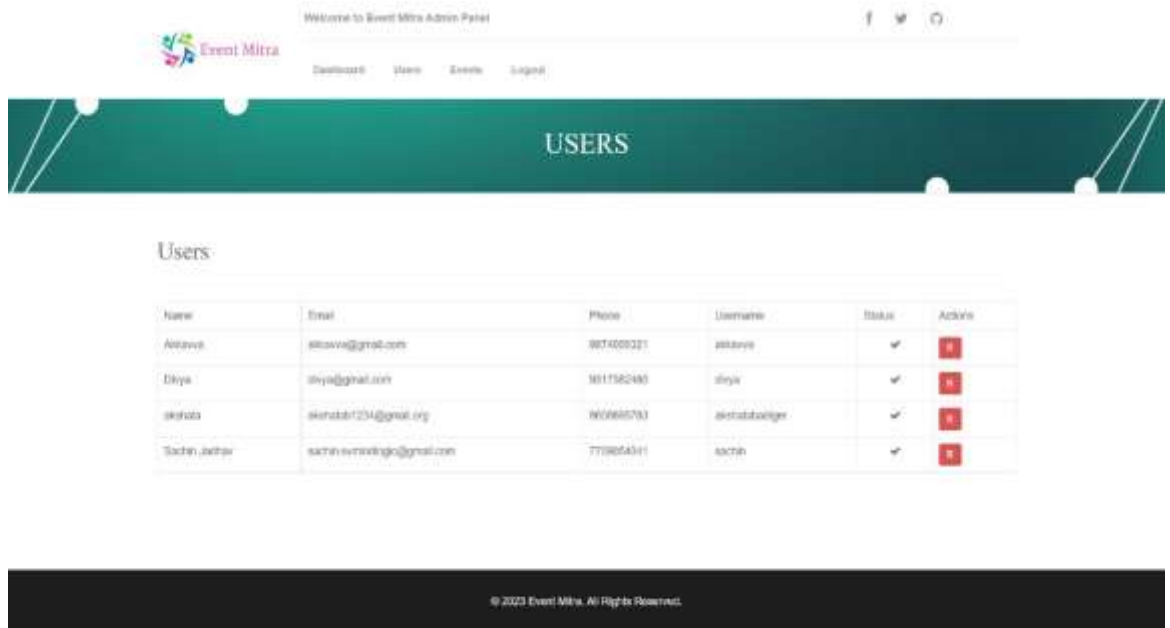
Fig.:3.1.7.3

Publisher Login

This screenshot shows the Event Mitra website's login page. The layout is consistent with the registration page, featuring the same navigation bar and green "LOGIN" banner. The "Login" form includes a "User Name" field (filled with "div") and a "Password" field (filled with "*****"). A pink "LOGIN" button is located below the fields, with a link "Don't have account? Register here" underneath. The browser's address bar shows the URL "http://localhost:8080/eventmitra/login.php". The Windows taskbar at the bottom indicates the system time as 11:40 AM on 11/01/2020.

Fig.:3.1.7.4

Publisher Registration List











Welcome to Event Mitra Admin Panel

Event Mitra

Dashboard Users Events Logout

USERS

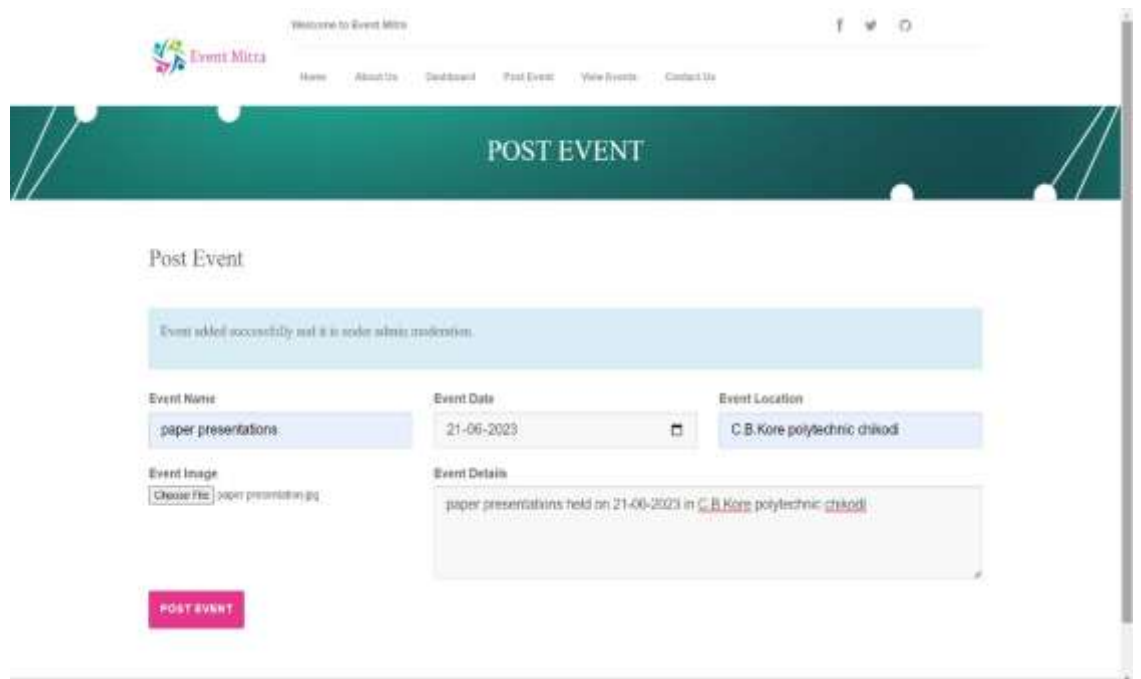
Users

| Name | Email | Phone | Username | Status | Action |
|--------------|-----------------------------|------------|----------------|--------|---|
| Ananya | ananyag@gmail.com | 987400321 | ananya | ✓ |   |
| Divya | divyag@gmail.com | 981782480 | divya | ✓ |   |
| Akshata | akshata1234@gmail.org | 989985783 | akshatabaliger | ✓ |   |
| Sachin Joshi | sachin.surwadekar@gmail.com | 7798654321 | sachin | ✓ |   |

© 2023 Event Mitra. All Rights Reserved.

Fig.:3.1.7.5

Create Event and Competition



Welcome to Event Mitra

Event Mitra

Home About Us Dashboard Post Event View Events Contact Us

POST EVENT


Post Event

Event added successfully and it is under admin's confirmation.

Event Name: paper presentations

Event Date: 21-06-2023

Event Location: C.B.Kore polytechnic chikodi

Event Image:  paper presentations.jpg

Event Details: paper presentations held on 21-06-2023 in C.B.Kore polytechnic chikodi

POST EVENT

Fig.:3.1.7.6

Posted Events

Welcome to Event Mitra

Home About Us Dashboard Post Event View Events Contact Us

POSTED EVENTS

Posted Events

| ID | Image | Event Name | Event Date | Event Location | Event Details | Status | Registration |
|----|-------|---------------------|------------|-------------------------------|---|--------|--------------------------|
| 5 | | paper presentations | 21/06/2023 | C.B.Kore polytechnic, chikodi | paper presentations held on 21-06-2023, Venue: C.B.Kore polytechnic chikodi | Active | Register |

© 2023 Event Mitra. All Rights Reserved.

Fig.:3.1.7.7

Manage and View Events

Welcome to Event Mitra Admin Panel

Dashboard Users Events Logout

EVENTS

Events

| Event Name | Event Date | Event Location | Event Details | Added By | Phone | Status | Actions |
|--------------------------|------------|-------------------------------|---|---------------|------------|------------------------------------|----------------------|
| essay competition | 27/06/2023 | C.B.Kore polytechnic, chikodi | Seminar Hall | Deepa | 0017362460 | ● | Edit |
| paper presentations | 21/06/2023 | C.B.Kore polytechnic, chikodi | paper presentations held on 21/06/2023, Venue: C.B.Kore polytechnic chikodi | Abhishek | 007403301 | ✓ | Edit |
| Chandigarh Half Marathon | 16/07/2023 | Chandigarh | 21KM (Half Marathon) 10.4KM 5KM 2KM Fun Run at Chandigarh 530000 | Sachin Jethwa | 7798504841 | ✓ | Edit |

© 2023 Event Mitra. All Rights Reserved.

Fig.:3.1.7.8

User Event List



Fig.:3.1.7.9

Apply for events



Fig.:3.1.7.10

View Applied Participants List

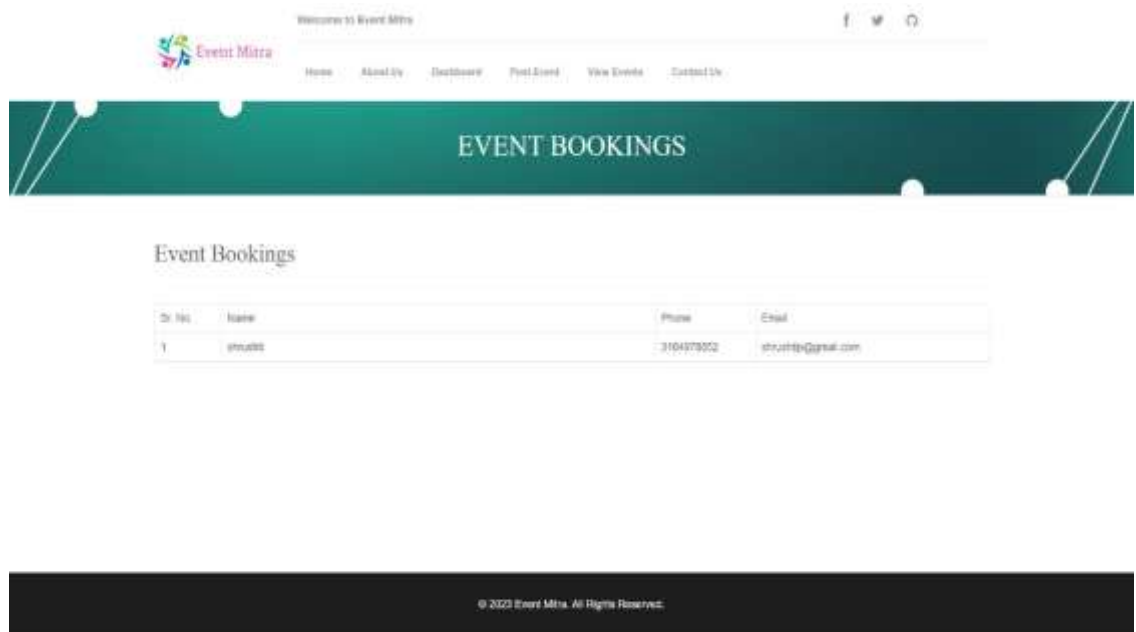


Fig.:3.1.7.11

3.1.8 Fabrication:

- Admin login: -

In admin login, first we have collected the requirements & after collecting requirements we analysed the needed requirements. Then we are designing the admin login form. After designing we are validating the form with filling the required details of each field with proper information. After we performed the database connection. Then, after completing all the steps testing will take place.

- Publisher registration: -

In publisher registration, first we have collected requirements, after that we analysed the requirements for this form. After that we take next step to design the form. Then we have validated the form with filling some details, after form validated correctly, then we went for database connection. Then we have tested the form with comparing each step of the form.

- Publisher Login: -

In publisher login form, we have collected requirements, after collecting requirements we analysed the requirements for this form. Then we have designed the form. Then we have validated the form with filling some details like username & password, after form validated correctly, then we went for database connection.

Then we have tested the form with comparing each step of the form.
- View publisher registration list: -

In view publisher registration list, we have collected requirements, after collecting requirements we analysed the requirements for this. Then we have designed this page. Then we have validated the form. Then we went for database connection.

Then we have tested the form with comparing each step of the form.
- Create events and competition: -

In this we have created the events, so for this first we have collected the requirements then we analysed the requirements. After that we have designed the page. After that we validated form with each field with filling details. Then performed the database connection. After all this we tested the page comparing with each step of the page.
- Manage and view events: -

In this admin is going to view events, & he should managing all the sub modules. For execution of this page, we have collected the requirements & analysed the requirements. Then we designed the page. After designing we have validated this page with entering some details. Then we performed the database connection. After this testing will take place.
- User view events list: -

In this user can view the created events. In this first we have collected the requirements then we analysed the requirements. After that we have designed the page. After that we validated form with each field with filling details. Then performed the database connection. After all this we tested the page comparing with each step of the page.

- Apply for events: -

In this user can apply for events with entering his correct details. For this we have collected requirements, after collecting requirements we analysed the requirements for this. Then we have designed the page. Then we have validated this page. After form validated correctly, then we went for database connection. Then we have tested the form with comparing each step of the page.

- View applied participants list: -

In this publisher can view the applied participants list for their events. For this we have collected requirements, after collecting requirements we analysed the requirements for this. Then we have designed the page. Then we have validated this page. After form validated correctly, then we went for database connection.

Then we have tested the form with comparing each step of the page.

System Architecture:

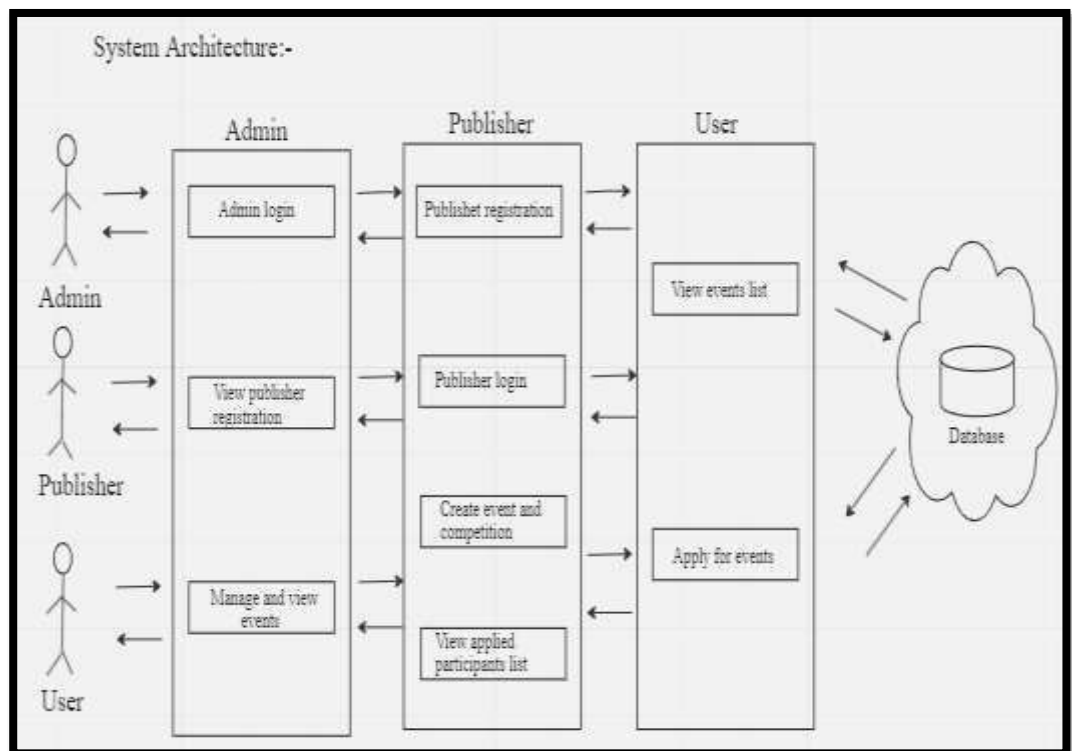


Fig.:3.1.8.1

CHAPTER 4

4.1 TEST AND VALIDATION

4.1.1 Test Plan:

- Software testing can be stated as the process of verifying and validating whether a software or application is bug-free, meets the technical requirements as guided by its design and development, and meets the user requirements effectively and efficiently by handling all the exceptional and boundary cases.

Software testing can be divided into two steps:

Verification: it refers to the set of tasks that ensure that the software correctly implements a specific function.

Validation: it refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements.

- Testing Types Manual Testing: -

Manual Testing is a kind of software testing in which a software tester develops and executes the test cases without using any automated testing tools. The main objective of manual testing is to detect the issues, bugs, and defects of a software application.

- Unit Testing: -

Unit testing is a software development process in which the smallest testable parts of an application, are called units. The main objective of unit testing is to isolate written code to test and determine if it works as intended. Unit testing is an important step in the development process. If done correctly, unit tests can detect early flaws in code which may be more difficult to find in later testing stages.

- **Integration Testing: -**

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements.

- **White Box Testing: -**

The technique of testing in which the tester is aware of the internal workings of the product, has access to its source code, and is conducted by making sure that all internal operations are performed according to the specifications is known as white box testing.

- **Black Box Testing: -**

The technique of testing in which the tester doesn't have access to the source code of the software and is conducted at the software interface without any concern with the internal logical structure of the software is known as black-box testing.

4.1.2 Test Approach

Admin Login

| Test Case Id | Test Description | Test Steps | Test Data | Expected Result | Actual Result | Status |
|--------------|-----------------------------------|-----------------------------|--------------------------------------|--------------------------|---------------|--------|
| TC01 | Check Admin Login with Valid Data | Enter Username And Password | Username = admin Password = admin | Admin Login Successfully | As Expected | Pass |

Table.:4.1.2.1

Publisher registration

| Test Case Id | Test Description | Test Steps | Test Data | Expected Result | Actual Result | Status |
|--------------|--|--|--|-----------------------------|---------------|--------|
| TC02 | Add Publisher registration with Valid Data | Enter Name, Phone Number, Email, Username and password | Name= Diya Phone number= 1593782465 Email =diyap@gmail.com Username = diya21 Password = patil21 | Thanks for the Registration | As Expected | Pass |

Table.:4.1.2.2

Publisher login

| Test Case Id | Test Description | Test Steps | Test Data | Expected Result | Actual Result | Status |
|--------------|---------------------------------------|-----------------------------|---|------------------------------|---------------|--------|
| TC03 | Check Publisher Login with Valid Data | Enter Username And Password | Username = diya21 Password = patil21 | Publisher Login Successfully | As Expected | Pass |

Table.:4.1.2.3

View publisher registration list

| Test Case Id | Test Description | Test Steps | Test Data | Expected Result | Actual Result | Status |
|--------------|--|--------------|------------------------|----------------------------------|---------------|--------|
| TC04 | Check view publisher registration list with Valid Data | Check Status | Activate or inactivate | User Status Changed Successfully | As Expected | Pass |

Table.:4.1.2.4

Create events and competition

| Test Case Id | Test Description | Test Steps | Test Data | Expected Result | Actual Result | Status |
|--------------|---|---|--|---|---------------|--------|
| TC05 | Add Create events and competition with Valid Data | Enter Event name, Event Date, Event Location, Event Image, Event Details | Event name= paper presentation Event date= 21-07-2023 Event location= C.B. Kore polytechnic Chikodi Event image= paperpresentation.jpg Event details= Seminar-hall | Event added successfully and it is under admin moderation | As Expected | Pass |

Table.:4.1.2.5

Manage and view events

| Test Case Id | Test Description | Test Steps | Test Data | Expected Result | Actual Result | Status |
|--------------|---|--------------|------------------------|-----------------------------------|---------------|--------|
| TC06 | Check view Manage and view events with Valid Data | Check Status | Activate or inactivate | Event Status Changed Successfully | As Expected | Pass |

Table.:4.1.2.6

Apply for events

| Test Case Id | Test Description | Test Steps | Test Data | Expected Result | Actual Result | Status |
|--------------|--------------------------------------|---------------------------------|---|--------------------|---------------|--------|
| TC07 | Add Apply for events with Valid Data | Enter Name, Phone Number, Email | Name= vidya Phone number= 1593765465 Email =vidyap@gmail.com | Booking successful | As Expected | Pass |

Table.:4.1.2.7

4.1.3 Features Tested:

Admin Login

Publisher

Registration

Publisher Login

View publisher registration

Create event and Competition

Manage and view events

User view events list

Apply for events

View applied participants list

4.1.4 Features Not Tested:

We have not tested on cloud server because we need to purchase cloud server.

4.1.5 Finding:

- Admin login

Empty Username & password: - If admin doesn't provide any username & password in username & password field, we will receive a pop-up message as please enter username & password, and we expected the same result, hence the test is passed.

- Publisher registration

Numeric Name: - If publisher provide numbers and special character in the name field, we will receive a pop-up message as name should not be numeric, and we expected the same result, hence the test is passed.

Empty Email: - Publisher doesn't entered email address in the field, we will receive a pop-up message as please enter valid email address, and we expected the same result, hence the test is passed.

- Publisher login

Enter Username & password: - here publisher need to enter the valid username and password. Otherwise, it will give pop-up message as please enter valid username & password, and we expected the same result, hence the test is passed.

View publisher registration list

- Status Activation: - Admin uses the status field to active and inactivate in View publisher registration list. When admin makes publishers status field active, then only publishers can easily login to their account.

- Create events and competition

Empty Name: -If publisher doesn't provide any event name in the event name field, we will receive a pop-up message as please enter event name, and we expected the same result, hence the test is passed.

Passed out date: -If publishers enter the passed-out date in the date field, we receive the pop-up message as Value must be today's date or later and we expected the same result, hence the test is passed.

- Manage and view events

Status Deactivation: - Admin is managing the events list, when admin make the event status active then it is visible to all users.

- Apply for events

Empty Phone Number: - If user doesn't provide any phone number in the phone number field, we will receive a pop-up message as please enter phone number, and we expected the same result, hence the test is passed.

4.1.6 Inference:

- Admin login

Admin login is really important in our project because admin manages everything related publishers and users. Publishers also plays an important role in Event Mitra. Publisher is dependent on admin for event publication.

- Publisher registration

Publisher must register with all the required fields in the form, then only he can login. Admin can view publisher registration list only of any publisher's registration.

- Publisher login

Publisher login is totally dependent on publisher registration. Then only their username and password are valid, and will be allowed to login.

- View publisher registration list

This list displays all the publishers list who have registered in Event Mitra. This list is dependent on publishers' registration.

Create events and competition

In this the publisher will create an event by providing all required details. This create event and competition is dependent on user events page and manage and view events page.

- Manage and view events

This page is only visible and handled by admin. Admin can activate or inactivate the event or can delete event. This is depended on create event and competition page and events list.

- User view events list

This page is viewed users, they can see all the events that are published and this dependent on create event and coopetition page.

Introduction: -

Event Mitra is an excellent choice for event planning. This website is useful for anyone who wish to run whatever sort of programme they desire. It may be used in many locations such as society, schools/colleges, and even event management organisations.

Proposed System: -

We will implement your project name. The purpose of this project is to create “EVENT MITRA”. In this project publisher wants to publish their events and user will take the registration for the event competition. Publisher needs to register and create events. After creating the events any user can see events list and participate. Admin can get the publisher registration list and created events. To implement this project, we will use full stack languages like PHP, HTML, CSS, Bootstrap, JavaScript, MySQL.

CHAPTER-5

5.1 BUSINESS ASPECTS

5.1.1 Briefly describe the market and economic outlook of the caption project for the industry

Events create opportunities for people to connect with an area, spend time together, celebrate and experience the diversity of cultures and foster creativity and innovation. They allow a community to come alive and provide an opportunity for a destination to showcase its tourism experience and increase economic activity. Events contribute significantly to community building, lifestyle and leisure enhancement, cultural development, volunteer participation, fundraising and economic development. Most importantly, events create a sense of fun and vibrancy, resulting in a strong sense of community connectivity, pride and a sense of place.

The purpose of this project is to create “Event Mitra”. In this project the publisher wants to publish their events and user will take the registration for the event competition. Publisher needs to register and create events. After creating the events any user can see events list and participate. Admin can get the publisher registration list and created events list. To implement this project, we will use full stack languages like PHP, HTML, CSS, Bootstrap, JavaScript, MySQL.

5.1.2 Highlight the novel features of Project/Service

Reduce the time of user.

Reduce the manual work.

Easy to use.

Access from any location.

Easily event registration.

No wastage of money.

5.1.3 How does a product/service fit into a competitive landscape

The product or service's fit into the competitive landscape depends on its unique value proposition and how it differentiates itself from its competitors. It is important to analyse the strengths and weaknesses of both the product/service and its competitors to identify opportunities for growth and potential threats.

5.1.4 Who are the possible capital project clients/customer

Event Organizers: Event Mitra could target professional event organizers who plan and execute events of different scales, such as corporate conferences, trade shows, concerts, weddings, or sports events. These organizers may require assistance with event management, ticketing, registration, venue selection, logistics, marketing, and other related services.

Corporate Companies: Large corporations often organize their own events, including product launches, seminars, training programs, or annual conferences. Event Mitra could cater to these companies by offering end-to-end event management solutions to handle the planning, logistics, branding, and attendee engagement.

Non-profit Organizations: Non-profit organizations frequently hold fundraising galas, charity events, awareness campaigns, and community gatherings. Event Mitra can provide its services to assist these organizations in managing their events effectively, optimizing ticket sales, and enhancing donor engagement.

Educational Institutions: Schools, colleges, and universities regularly organize academic conferences, student orientations, cultural festivals, and sports events. Event Mitra could offer its platform to simplify event planning, manage registrations, and streamline communication among participants.

Government Agencies: Government bodies often host conferences, seminars, public consultations, and cultural events. Event Mitra could collaborate with government agencies to provide event management tools and services, ensuring smooth execution and improved attendee experiences.

Project Advantage:-

Reduce the time of user.

Reduce the manual work.

Easy to use.

Access from any location.

Easily event registration.

No wastage of money.

5.2 FINANCIAL CONSIDERATIONS

5.2.1 Capstone project budget:

The total estimated cost of this project 46,000

5.2.2 Cost capstone projections needed for either for profit/non-profit options.

| Title | Cost (Rupees) |
|------------------|----------------------|
| Estimated Cost | ₹46,000 |
| Labour Cost | ₹25,200 |
| Cost of material | ₹8,800 |
| Net Profit | ₹12,000 |

Table.:5.2.2.1

5.3 CONCLUSION AND RECOMMENDATIONS

5.3.1 Describe state of complement of caption project

The publisher who wants to develop and publish an event on this website must first register using the register form. After registration using the registered username and password, they may login and create the event, but participants will not be able to access it until it is authorised by admin. Any user, once allowed, can participate in any event. The administrator of this site has pre-defined a username and password for secure access to the relevant fields. Only the administrator may login; users and participants do not need to register or login.

The posts on this website are published by the publisher. The publishers design the event with a good description and photos for explanation of the rules and regulations, place, dress, fees, and so on. Following that, attendees can apply for activities in which they are interested by completing the necessary information and registering.

5.3.2 Future Work: -

We can live the project to any institute.

We can develop the mobile application.

CODE

Index:-

```
<!DOCTYPE HTML>
<html class="no-js">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title>Event Mitra</title>
<meta name="viewport" content="width=device-width, user-scalable=no, initial-
scale=1.0, minimum-scale=1.0, maximum-scale=1.0"/>
<meta name="format-detection" content="telephone=no"/>
<link href="css/bootstrap.css" rel="stylesheet" type="text/css"/>
<link href="css/style.css" rel="stylesheet" type="text/css"/>
<link      href="plugins/sequence/css/sequence.html"      rel="stylesheet"
type="text/css"/>
<link      href="plugins/prettyphoto/css/prettyPhoto.css"      rel="stylesheet"
type="text/css"/>
<link      href="plugins/owl-carousel/css/owl.carousel.css"      rel="stylesheet"
type="text/css"/>
<link      href="plugins/owl-carousel/css/owl.theme.css"      rel="stylesheet"
type="text/css"/>
<link rel="stylesheet" type="text/css" href="plugins/rs-plugin/css/settings.css"
media="screen" />
<!--[if lte IE 8]><link rel="stylesheet" type="text/css" href="css/ie8.css"
media="screen" /><![endif]-->
<link class="alt" href="colors/pink.css" rel="stylesheet" type="text/css"/>
<script src="js/modernizr.js"></script>
<?php
require_once "eventhelper.php";
$helper = new EventHelper();
?>
</head>
```

```

<body class="" style="background-image:
url('images/backgrounds/images/img3.jpg'); background-repeat: no-repeat;
background-size: cover;">
<!--[if lt IE 7]>
    <p class="chromeiframe">You are using an outdated browser. <a
href="http://browsehappy.com/">Upgrade your browser today</a> or <a
href="http://www.google.com/chromeiframe/?redirect=true">install Google
Chrome Frame</a> to better experience this site.</p>
<![endif]-->
<!-- Start Body Container -->
<div class="body footer-style2 header-style3">
    <!-- Start Header -->

    <?php
        require_once "header.php";
    ?>

    <!-- End Header -->

    <!-- Start Content -->
    <div class="main" role="main">
        <div id="content" class="content full">
            <div class="rev-slider-container">
                <div class="tp-banner-container">
                    <div class="tp-banner" >
                        <ul>
                            <!-- SLIDE -->
                            <li data-delay="4000" data-masterspeed="600" data-slotamount="7"
data-transition="scaledownfromtop">
                                <!-- MAIN IMAGE -->
                                
                            </li>
                            <li data-delay="4000" data-masterspeed="600" data-slotamount="7"
data-transition="scaledownfromtop">
                                <!-- MAIN IMAGE -->
                                
                            </li>

```

```

        </ul>
    </div>
</div>
<div class="featured-row dgray-color margin-20">
    <div class="container">
        <div class="row">
            <div class="col-md-12">
                <h2>Welcome to Event Mitra.</h2>
                <a href="aboutus.php" class="btn btn-primary">Know more</a> </div>
            </div>
        </div>
    </div>
<div class="padding-tb75">
    <div class="container text-align-center">

        <div class="row">
            <div class="col-md-4" data-appear-animation="bounceIn" data-appear-
animation-delay="5">
                <section class="features text-align-center">
                    <i class="ic-lg fa fa-eye fa-2x"></i>
                    <h3 class="short">Register</h3>
                    <hr class="sm" />
                    <p class="md">In this site the publisher who want to create and
publish the event have to register under register form. After registering using
registerd username and password they can login and create the event and unless the
event is approved by admin participants can't see the event. once approved, any
user can participate for any events.</p>
                </section>
            </div>
            <div class="col-md-4" data-appear-animation="bounceIn" data-appear-
animation-delay="25">
                <section class="features text-align-center">
                    <i class="ic-lg fa fa-headphones fa-2x"></i>

```

```

        <h3 class="short">Login</h3>
        <hr class="sm" />
        <p class="md">In this site admin have pre-defined username and
password for secure access to the important fields. Other than admin only publisher
can login, user or participants don't have to register or login.</p>
    </section>
</div>
<div class="col-md-4" data-appear-animation="bounceIn" data-appear-
animation-delay="75">
    <section class="features text-align-center">
        <i class="ic-lg fa fa-thumbs-o-up fa-2x"></i>
        <h3 class="short">Post </h3>
        <hr class="sm" />
        <p class="md">In this site the post are posted by publisher. The
publisher's create the event with proper description and images of their event for
clarification about rules and regulation, venue, attire, fees, etc. After that participants
can apply for the events which they are interested in and can apply for events by
providing their required data and register.</p>
    </section>
</div>
</div>
</div>
</div>
</div>
</div>
<?php
require_once "footer.php";
?>
</div>
<!-- End Body Container -->
<script src="js/jquery-latest.min.js"></script> <!-- JQuery Library Call -->
<script src="plugins/prettyphoto/js/prettyphoto.js"></script>
<script src="plugins/owl-carousel/js/owl.carousel.min.js"></script>
<script src="plugins/page-scroller/jquery.pagedscroller.js"></script>

```

```

<script src="js/helper-plugins.js"></script> <!-- Plugins -->
<script src="js/bootstrap.js"></script> <!-- UI -->
<script src="js/init.js"></script> <!-- All Scripts -->
<script src="plugins/rs-plugin/js/jquery.themepunch.plugins.min.js"></script>
<script src="plugins/rs-plugin/js/jquery.themepunch.revolution.min.js"></script>
<script src="js/revolution-slider-init.js"></script> <!-- Revolutions Slider
Intialization -->
<!-- Preloader -->
<script type="text/javascript">
    <![CDATA[
        $(window).load(function() { // makes sure the whole site is loaded
            $("#status").fadeOut(); // will first fade out the loading animation
            $("#preloader").delay(350).fadeOut("slow"); // will fade out the white DIV that
            covers the website.
        });
    </script>
</script>
<!-- End Js -->
</body>
</html>

```

Admin :-

```

<!DOCTYPE HTML>

<html class="no-js">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

<title>Event Mitra</title>

<meta name="viewport" content="width=device-width, user-scalable=no, initial-
scale=1.0, minimum-scale=1.0, maximum-scale=1.0"/>

<meta name="format-detection" content="telephone=no"/>

<link href="../css/bootstrap.css" rel="stylesheet" type="text/css"/>

```

```

<link href="../../css/style.css" rel="stylesheet" type="text/css"/>

<link href="../../plugins/sequence/css/sequence.html" rel="stylesheet"
type="text/css"/>

<link href="../../plugins/prettyphoto/css/prettyPhoto.css" rel="stylesheet"
type="text/css"/>

<link href="../../plugins/owl-carousel/css/owl.carousel.css" rel="stylesheet"
type="text/css"/>

<link href="../../plugins/owl-carousel/css/owl.theme.css" rel="stylesheet"
type="text/css"/>

<link rel="stylesheet" type="text/css" href="../../plugins/rs-plugin/css/settings.css"
media="screen" />

<!--[if lte IE 8]><link rel="stylesheet" type="text/css" href="../../css/ie8.css"
media="screen" /><![endif]-->

<link class="alt" href="../../colors/pink.css" rel="stylesheet" type="text/css"/>

<script src="../../js/modernizr.js"></script>

<script type="text/javascript">

    function validate_form()

    {

        var username = document.getElementById("username").value;

        var password = document.getElementById("password").value;

        if(username=="")

        {

            alert("Please Enter User Name.");

            return false;

        }

        else if(password=="")

        {

            alert("Please Enter Password.");

            return false;

        }

    }

}

```



```

    }

}

</script>

<?php
require_once "adminhelper.php";

$helper = new AdminHelper();

?>

</head>

<body class="" >

<!--[if lt IE 7]>

    <p class="chromeiframe">You are using an outdated browser. <a
href="http://browsehappy.com/">Upgrade your browser today</a> or <a
href="http://www.google.com/chromeiframe/?redirect=true">install Google
Chrome Frame</a> to better experience this site.</p>

<![endif]-->

<!-- Preloader -->

<div id="preloader">

    <div id="status"></div>

</div>

<!-- Start Body Container -->

<div class="body footer-style2 header-style3">

    <!-- Start Header -->

    <?php
        require_once "header.php";
    ?>

    <!-- End Header -->

    <!-- Start Content -->

```

```

<div class="main" role="main">

  <div id="content" class="content page-content full">

    <header class="page-header flexible parallax text-align-center parallax-
overlay" style="background-image:url(..images/img4.jpg)">

      <section>

        <div class="container">

          <div class="row">

            <div class="col-md-12">

              <h1>Login</h1>

            </div>

          </div>

        </div>

      </section>

    </header>

    <div class="container">

      <div class="row">

        <div class="col-md-4 col-md-offset-4">

          <h2><strong>Login</strong></h2>

          <hr/>

          <?php
            if($_GET['error'])
            {
              ?>

              <div class="alert alert-info" role="alert">

                <h3 style="text-transform: none;">Login Failed! Please enter valid
details.</h3>

              </div>

            <?php

```

```

    }
    ?>

    <form method="post" action="checklogin.php" onsubmit="return
validate_form();">

        <div class="row">

            <div class="form-group">

                <div class="col-md-12">

                    <label>User Name</label>

                    <input type="text" id="username" name="username" class="form-
control input-lg" placeholder="Username">

                </div>

            </div>

        </div>

        <div class="row">

            <div class="form-group">

                <div class="col-md-12">

                    <label>Password</label>

                    <input type="password" id="password" name="password"
class="form-control input-lg" placeholder="Password">

                </div>

            </div>

        </div>

        <div class="row">

            <div class="form-group">

                <div class="col-md-12">

                    <input type="submit" name="submit" class="btn btn-primary btn-lg"
value="Login">

                </div>

            </div>

        </div>
    </form>

```

```

        </div>

    </div>

</form>

</div>

<!-- Start Sidebar -->

<aside class="col-md-3 sidebar right-sidebar">

</aside>

</div>

</div>

</div>

</div>

</div>

<?php
    require_once "footer.php";
?>

</div>

<!-- End Body Container -->

<script src="../../js/jquery-latest.min.js"></script> <!-- JQuery Library Call -->

<script src="../../plugins/prettyphoto/js/prettyphoto.js"></script>

<script src="../../plugins/prettyphoto/js/prettyphoto.js"></script>

<script src="../../plugins/owl-carousel/js/owl.carousel.min.js"></script>

<script src="../../plugins/page-scroller/jquery.pagescroller.js"></script>

<script src="../../js/helper-plugins.js"></script> <!-- Plugins -->

<script src="../../js/bootstrap.js"></script> <!-- UI -->

<script src="../../js/init.js"></script> <!-- All Scripts -->

<!-- Preloader -->

<script type="text/javascript">

    //<![CDATA[

```

```

$(window).load(function() { // makes sure the whole site is loaded

$("#status").fadeOut(); // will first fade out the loading animation

$("#preloader").delay(350).fadeOut("slow"); // will fade out the white DIV
that covers the website.

});

//]]>
</script>
<!-- End Js -->
</body>
</html>

```

Publisher

Publisher registration:-

```

<!DOCTYPE HTML>

<html class="no-js">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

<title>Event Mitra</title>

<meta name="viewport" content="width=device-width, user-scalable=no, initial-
scale=1.0, minimum-scale=1.0, maximum-scale=1.0"/>

<meta name="format-detection" content="telephone=no"/>

<link href="css/bootstrap.css" rel="stylesheet" type="text/css"/>

<link href="css/style.css" rel="stylesheet" type="text/css"/>

<link href="plugins/sequence/css/sequence.html" rel="stylesheet"
type="text/css"/>

<link href="plugins/prettyphoto/css/prettyPhoto.css" rel="stylesheet"
type="text/css"/>

<link href="plugins/owl-carousel/css/owl.carousel.css" rel="stylesheet"
type="text/css"/>

```

```

<link href="plugins/owl-carousel/css/owl.theme.css" rel="stylesheet"
type="text/css"/>

<link rel="stylesheet" type="text/css" href="plugins/rs-plugin/css/settings.css"
media="screen" />

<!--[if lte IE 8]><link rel="stylesheet" type="text/css" href="css/ie8.css"
media="screen" /><![endif]-->

<link class="alt" href="colors/pink.css" rel="stylesheet" type="text/css"/>

<script src="js/modernizr.js"></script>

<script type="text/javascript">

    function validate_form()

    {

        var name      = document.getElementById("name").value;
        var phone      = document.getElementById("phone").value;
        var email      = document.getElementById("email").value;
        var username    = document.getElementById("username").value;
        var passwordInput = document.getElementById("password");
        var password = passwordInput.value;
        var validchar = /^[A-Z a-z]+$;/

        //name

        if (name.trim() === "") {

            alert("Name cannot be empty.");

            return false;

        }

        var repeatingCharsRegex = /([a-zA-Z])\1{2}/;

        if (repeatingCharsRegex.test(name)) {

            alert("Name should not have repeating alphabets more than 2 times.");

            return false;

        }

    }


```

```

var specialCharsRegex = /[!@#$$%^&*(),.?":{}|<>]/;
if (specialCharsRegex.test(name)) {
    alert("Name should not contain special characters.");
    return false;
}
var specialCharsRegex = /^[a-zA-Z\s]/;
if (specialCharsRegex.test(name)) {
    alert("Name should only contain alphabets and spaces.");
    return false;
}
//phonenummer
else if(phone=="")
{
    alert("Please Enter Your Phone Number.");
    return false;
}
else if(isNaN(phone))
{
    alert("Phone Number should be numeric.");
    return false;
}
else if(checkInternationalPhone(phone)==false)
{
    alert("Please Enter a Valid Phone Number");
    return false;
}
//email

```

```

if (email.trim() === "") {
    alert("email cannot be empty.");
    return false;
}

var emailRegex = /^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[cC][oO][mM]$/;
if (!emailRegex.test(email)) {
    alert("Invalid email format.");
    return false;
}

var repeatingCharsRegex = /^[A-Za-z0-9._%+-]\1+@/;
if (repeatingCharsRegex.test(email)) {
    alert("Email should not have repeating characters or alphabets before and
after the @ symbol.");
    return false;
}

var emailRegex = /^[A-Za-z0-9._%+-]+@gmail\.com$/;
if (!emailRegex.test(email)) {
    alert("Invalid Gmail address.");
    return false;
}

else if(validateEmail(email))
{
    alert("Please Enter Valid Email Address.");
    return false;
}

//username

if (username.trim() === "") {

```



```

        alert("Username cannot be empty.");

        return false;
    }

    var numericRegex = /[0-9]/;

    if (numericRegex.test(username)) {

        alert("Username should not contain numeric characters.");

        return false;
    }

    var specialCharsRegex = /^[^a-zA-Z\s]/;

    if (specialCharsRegex.test(username)) {

        alert("Username should only contain alphabets and spaces.");

        return false;
    }

    var repeatingCharsRegex = /([a-zA-Z])\1{2}/;

    if (repeatingCharsRegex.test(username)) {

        alert("Username should not have repeating characters or alphabets more than
2 times.");

        return false;
    }

    //password

    if (password.trim() === "") {

        alert("Password cannot be empty.");

        return false;
    }

    // Check password against criteria

    var uppercaseRegex = /[A-Z]/;

```

```

var lowercaseRegex = /[a-z]/;
var numericRegex = /[0-9]/;
var specialCharRegex = /[!@#$%^&*()]/;

if (
    password.length < 8 ||
    !uppercaseRegex.test(password) ||
    !lowercaseRegex.test(password) ||
    !numericRegex.test(password) ||
    !specialCharRegex.test(password)
) {
    alert(
        "Password should be at least 8 characters long and include at least one
        uppercase letter, one lowercase letter, one numeric digit, and one
        special character."
    );
    return false;
}

function validateEmail(email)
{
    var atpos = email.indexOf("@"); // The indexOf() method returns the position
    of the first occurrence of a specified value in a string. // Default value of start is 0

    //alert(atpos);

    var dotpos = email.lastIndexOf("."); // The lastIndexOf() method returns the
    position of the last occurrence of a specified value in a string. // Default value of
    start is 0

    //alert(dotpos);

    if((atpos<1) || (dotpos<(atpos+2)) || (dotpos+2>=email.length))
    {

```

```

        return true;
    }
    else
    {
        return false;
    }
}

// Declaring required variables
var digits = "0123456789";

// non-digit characters which are allowed in phone numbers
var phoneNumberDelimiters = "() - ";

// characters which are allowed in international phone numbers
// (a leading + is OK)
var validWorldPhoneChars = phoneNumberDelimiters + "+";

// Minimum no of digits in an international phone no.
var minDigitsInIPhoneNumber = 10;

function isInteger(s)
{
    var i;
    for (i = 0; i < s.length; i++)
    {
        // Check that current character is number.
        var c = s.charAt(i);
        if (((c < "0") || (c > "9")))) return false;
    }

    // All characters are numbers.

    return true;
}

```

```

function trim(s)
{
    var i;

    var returnString = "";

    // Search through string's characters one by one.

    // If character is not a whitespace, append to returnString.

    for (i = 0; i < s.length; i++)
    {
        // Check that current character isn't whitespace.

        var c = s.charAt(i);

        if (c != " ") returnString += c;
    }

    return returnString;
}

function stripCharsInBag(s, bag)
{
    var i;

    var returnString = "";

    // Search through string's characters one by one.

    // If character is not in bag, append to returnString.

    for (i = 0; i < s.length; i++)
    {
        // Check that current character isn't whitespace.

        var c = s.charAt(i);

        if (bag.indexOf(c) == -1) returnString += c;
    }

    return returnString;
}

function checkInternationalPhone(strPhone){

```

```

        var bracket=3;

        strPhone=trim(strPhone);

        if(strPhone.indexOf("+")>1) return false;

        if(strPhone.indexOf("-")!=-1)bracket=bracket+1;

        if(strPhone.indexOf("(")!=-1 && strPhone.indexOf(">bracket)return
false;

        var brchr=strPhone.indexOf("(");

        if(strPhone.indexOf("(")!=-1 && strPhone.charAt(brchr+2)!=")")return false;

        if(strPhone.indexOf("(")==-1 && strPhone.indexOf(">")!=-1)return false;

        s=stripCharsInBag(strPhone,validWorldPhoneChars);

        return (isInteger(s) && s.length >= minDigitsInIPhoneNumber);

    }
</script>

<?php
require_once "eventhelper.php";

$helper = new EventHelper();

$msg = "";

if($_POST)
{
    $r = $helper->addUser();

    if($r)
    {
        $msg='<div class="alert alert-info" role="alert">

                <h3>Thanks for the registration.</h3>

            </div>';
    }
    else

```

```

    {
        $msg='<div class="alert alert-info" role="alert">

                <h3>Username Already Exists.</h3>

            </div>';
    }
}

?>

</head>

<body class="" >

<!--[if lt IE 7]>

    <p class="chromeframe">You are using an outdated browser. <a
href="http://browsehappy.com/">Upgrade your browser today</a> or <a
href="http://www.google.com/chromeframe/?redirect=true">install Google
Chrome Frame</a> to better experience this site.</p>

<![endif]-->

<!-- Preloader -->

<div id="preloader">

    <div id="status"></div>

</div>

<!-- Start Body Container -->

<div class="body footer-style2 header-style3">

    <!-- Start Header -->

    <?php
        require_once "header.php";
    ?>

    <!-- End Header -->

    <!-- Start Content -->

    <div class="main" role="main">

```

```

<div id="content" class="content page-content full">

    <header class="page-header flexible parallax text-align-center parallax-
overlay" style="background-image:url(images/img4.jpg)">

        <section>

            <div class="container">

                <div class="row">

                    <div class="col-md-12">

                        <h1>Register</h1>

                    </div>

                </div>

            </div>

        </section>

    </header>

    <div class="container">

        <div class="row">

            <div class="col-md-12">

                <h2><strong>Register</strong></h2>

                <hr/>

                <p><?php echo $msg;?></p>

                <form method="post" action="" onsubmit="return validate_form();"
enctype="multipart/form-data">

                    <div class="row">

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

                            <div class="form-group">

                                <label>Your Name</label>

                                <input type="text" id="name" name="name" class="form-control input-lg"
placeholder="Your Name"/>

                            </div>

```

```

</div>

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

  <div class="form-group">

    <label>Your Phone</label>

    <input type="text" id="phone" name="phone" maxlength="10"
class="form-control input-lg" placeholder="Your Phone"/>

  </div>

</div>

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

  <div class="form-group">

    <label>Your Email</label>

    <input type="text" id="email" name="email" class="form-
control input-lg" placeholder="Your Email"/>

  </div>

</div>

</div>

<div class="row">

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

    <div class="form-group">

      <label>User Name</label>

      <input type="text" id="username" name="username"
class="form-control input-lg" placeholder="Username"/>

    </div>

  </div>

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

    <div class="form-group">

      <label>Password</label>

```



```

        <input type="password" id="password" name="password"
class="form-control input-lg" placeholder="Password"/>

    </div>

</div>

</div>

<div class="row">

    <div class="form-group">

        <div class="col-md-12">

            <input type="submit" name="submit" class="btn btn-primary
btn-lg" value="Register now!"/>

        </div>

    </div>

</div>

</form>

<p>Have account <a href="login.php">Login Here</a></p>

</div>

<!-- Start Sidebar -->

<aside class="col-md-3 sidebar right-sidebar">

</aside>

</div>

</div>

</div>

</div>

<?php
    require_once "footer.php";

?>

</div>

<!-- End Body Container -->

```

```

<script src="js/jquery-latest.min.js"></script> <!-- JQuery Library Call -->

<script src="plugins/prettyphoto/js/prettyphoto.js"></script>

<script src="plugins/prettyphoto/js/prettyphoto.js"></script>

<script src="plugins/owl-carousel/js/owl.carousel.min.js"></script>

<script src="plugins/page-scroller/jquery.pagescroller.js"></script>

<script src="js/helper-plugins.js"></script> <!-- Plugins -->

<script src="js/bootstrap.js"></script> <!-- UI -->

<script src="js/init.js"></script> <!-- All Scripts -->

<!-- Preloader -->

<script type="text/javascript">

    //<![CDATA[

        $(window).load(function() { // makes sure the whole site is loaded

            $("#status").fadeOut(); // will first fade out the loading animation

            $("#preloader").delay(350).fadeOut("slow"); // will fade out the white DIV
            that covers the website.

            });

        //]]>

</script>

<!-- End Js -->

</body>

</html>

Publisher login:-

<!DOCTYPE HTML>

<html class="no-js">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

```

```

<title>Event Mitra</title>

<meta name="viewport" content="width=device-width, user-scalable=no, initial-
scale=1.0, minimum-scale=1.0, maximum-scale=1.0"/>

<meta name="format-detection" content="telephone=no"/>

<link href="css/bootstrap.css" rel="stylesheet" type="text/css"/>

<link href="css/style.css" rel="stylesheet" type="text/css"/>

<link href="plugins/sequence/css/sequence.html" rel="stylesheet"
type="text/css"/>

<link href="plugins/prettyphoto/css/prettyPhoto.css" rel="stylesheet"
type="text/css"/>

<link href="plugins/owl-carousel/css/owl.carousel.css" rel="stylesheet"
type="text/css"/>

<link href="plugins/owl-carousel/css/owl.theme.css" rel="stylesheet"
type="text/css"/>

<link rel="stylesheet" type="text/css" href="plugins/rs-plugin/css/settings.css"
media="screen" />

<!--[if lte IE 8]><link rel="stylesheet" type="text/css" href="css/ie8.css"
media="screen" /><![endif]-->

<link class="alt" href="colors/pink.css" rel="stylesheet" type="text/css"/>

<script src="js/modernizr.js"></script>

<script type="text/javascript">

    function validate_form()

    {

        var username = document.getElementById("username").value;

        var password = document.getElementById("password").value;

        if(username=="")

        {

            alert("Please Enter User Name.");

            return false;

        }

    }

```

```

        else if(password=="")
        {
            alert("Please Enter Password.");
            return false;
        }
    }
</script>

<?php
require_once "eventhelper.php";

$helper = new EventHelper();

?>

</head>

<body class="" >

<!--[if lt IE 7]>

    <p class="chromeframe">You are using an outdated browser. <a
href="http://browshappy.com/">Upgrade your browser today</a> or <a
href="http://www.google.com/chromeframe/?redirect=true">install Google
Chrome Frame</a> to better experience this site.</p>

<![endif]-->

<!-- Preloader -->

<div id="preloader">

    <div id="status"></div>

</div>

<!-- Start Body Container -->

<div class="body footer-style2 header-style3">

    <!-- Start Header -->

    <?php
        require_once "header.php";

```

```

?>

<!-- End Header -->

<!-- Start Content -->

<div class="main" role="main">

    <div id="content" class="content page-content full">

        <header class="page-header flexible parallax text-align-center parallax-
        overlay" style="background-image:url(images/img4.jpg)">

            <section>

                <div class="container">

                    <div class="row">

                        <div class="col-md-12">

                            <h1>Login</h1>

                        </div>

                    </div>

                </div>

            </section>

        </header>

        <div class="container">

            <div class="row">

                <div class="col-md-4 col-md-offset-4">

                    <h2><strong>Login</strong></h2>

                    <hr/>

                    <?php

                        if($_GET['error'])

                        {

                            ?>

                            <div class="alert alert-info" role="alert">

```

```
        <h3 style="text-transform: none;">Login Failed! Please enter valid
details.</h3>
```

```
    </div>
```

```
    <?php
```

```
    }
```

```
    ?>
```

```
    <form method="post" action="checkuser.php" onsubmit="return
validate_form();">
```

```
        <div class="row">
```

```
            <div class="form-group">
```

```
                <div class="col-md-12">
```

```
                    <label>User Name</label>
```

```
                    <input type="text" id="username" name="username" class="form-
control input-lg" placeholder="Username">
```

```
                </div>
```

```
            </div>
```

```
        </div>
```

```
        <div class="row">
```

```
            <div class="form-group">
```

```
                <div class="col-md-12">
```

```
                    <label>Password</label>
```

```
                    <input type="password" id="password" name="password"
class="form-control input-lg" placeholder="Password">
```

```
                </div>
```

```
            </div>
```

```
        </div>
```

```
        <div class="row">
```

```
            <div class="form-group">
```

```

        <div class="col-md-12">

            <input type="submit" name="submit" class="btn btn-primary btn-lg"
value="Login">

        </div>

    </div>

</div>

</form>

<p>Don't have account <a href="register.php">Register Here</a></p>

</div>

<!-- Start Sidebar -->

<aside class="col-md-3 sidebar right-sidebar">

</aside>

</div>

</div>

</div>

</div>

<?php
    require_once "footer.php";
?>

</div>

<!-- End Body Container -->

<script src="js/jquery-latest.min.js"></script> <!-- JQuery Library Call -->

<script src="plugins/prettyphoto/js/prettyphoto.js"></script>

<script src="plugins/prettyphoto/js/prettyphoto.js"></script>

<script src="plugins/owl-carousel/js/owl.carousel.min.js"></script>

<script src="plugins/page-scroller/jquery.pagescroller.js"></script>

<script src="js/helper-plugins.js"></script> <!-- Plugins -->

```

```

<script src="js/bootstrap.js"></script> <!-- UI -->

<script src="js/init.js"></script> <!-- All Scripts -->

<!-- Preloader -->

<script type="text/javascript">

    <![CDATA[

        $(window).load(function() { // makes sure the whole site is loaded

            $("#status").fadeOut(); // will first fade out the loading animation

            $("#preloader").delay(350).fadeOut("slow"); // will fade out the white DIV
            that covers the website.

            });

        <![>

    </script>

<!-- End Js -->

</body>

</html>

```


References and Bibliography

<https://reactjs.org/>

<https://github.com/>

<https://www.w3schools.com/REACT/DEFAULT.ASP>

<https://www.javatpoint.com/reactjs-tutorial>

<https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs>

Appendices

Chapter 1

| | |
|----------------------------------|-----|
| 1.1 Introduction | 1 |
| 1.2 Scope of the caption project | 1-3 |

Chapter 2

| | |
|---|-------|
| 2.1 Capstone project planning | |
| 2.1.1 Work breakdown structure | 4-5 |
| 2.1.2 Timeline developer-schedule | 5-8 |
| 2.1.3 Cost breakdown structure | 8-10 |
| 2.1.4 Capstone project risk assessment | 11 |
| 2.2 Requirements specification | |
| 2.2.1 Functional Requirements | 11-12 |
| 2.2.2 Non-functional (Quality attributes) | 12-13 |
| 2.2.3 User input | 13-14 |
| 2.2.4 Technical constraints | 14 |
| 2.3 Design specifications | |
| 2.3.1 Chosen system design | 15 |
| 2.3.2 Discussion of alternative design | 16-17 |
| 2.3.3 Detailed description of components/subsystems | 18 |
| 2.3.4 Components 1-n | 19-20 |

Chapter 3

| | |
|------------------------------|-------|
| 3.1 Approach and methodology | |
| 3.1.1 Discuss the technology | 21-23 |
| 3.1.2 Methodologies | 24 |
| 3.1.3 Use cases | 25 |
| 3.1.4 Programming | 26-27 |
| 3.1.5 Analysis | 28 |
| 3.1.6 Process design | 28-30 |
| 3.1.7 Product design | 31-36 |
| 3.1.8 Fabrication | 36-38 |

Chapter 4

| | |
|---------------------------|-------|
| 4.1 Test and validation | |
| 4.1.1 Test plan | 39-40 |
| 4.1.2 Test approach | 40-43 |
| 4.1.3 Features tested | 43 |
| 4.1.4 Features not tested | 43 |
| 4.1.5 Findings | 44-45 |
| 4.1.6 Inference | 45-46 |

Chapter 5

| | |
|---|-------|
| 5.1 Business aspects | |
| 5.1.1 The market and economic outlook of the project | 47 |
| 5.1.2 Features | 47 |
| 5.1.3 How does the project of service fit into the competitive landscape? | 48 |
| 5.1.4 Who want the possible client or customer? | 48-49 |
| 5.2 Financial considerations | |
| 5.2.1 Capstone project budget | 49 |

| | |
|--|-------|
| 5.2.2 Cost of capstone projection needed for either profit or nonprofit option | 49 |
| 5.3 Conclusion and recommendations | |
| 5.3.1 State of completion of capstone project | 50 |
| 5.3.2 Future work | 50 |
| Program Code | 51-80 |
| References | 81 |
| Appendices | 81-85 |

List of Tables:-

| Table number | Table name | Page number |
|---------------|----------------------------------|-------------|
| Table 2.1.3.1 | Labor cost of work | 9-10 |
| Table 2.1.3.2 | Final check | 10 |
| Table 4.1.2.1 | Admin login | 40 |
| Table 4.1.2.2 | Publisher Registration | 41 |
| Table 4.1.2.3 | Publisher Login | |
| Table 4.1.2.4 | View publisher registration list | |
| Table 4.1.2.5 | Create events and competition | 42 |
| Table 4.1.2.6 | Manage and view events | |
| Table 4.1.2.7 | Apply for events | 43 |
| Table 5.2.2.1 | Cost (Profit/non-profit) | 49 |

List of Figures:-

| Figure number | Figure name | Page number |
|---------------|--|-------------|
| Fig 2.3.1.1 | System Architecture | 15 |
| Fig 2.3.2.1 | Data flow diagram level 0 | 16 |
| Fig 2.3.2.2 | Data flow diagram level 1 Admin | |
| Fig 2.3.2.3 | Data flow diagram level 1 Publisher | 17 |
| Fig 2.3.2.4 | Data flow diagram level 1 User | |
| Fig 2.3.3.1 | Component diagram | 18 |
| Fig 3.1.2.1 | Agile methodology diagram | 24 |
| Fig 3.1.3.1 | Use case | 25 |
| Fig 3.1.6.1 | Table | 28 |
| Fig 3.1.6.2 | Admin | 29 |
| Fig 3.1.6.3 | Event | |
| Fig 3.1.6.4 | Event Booking | 30 |
| Fig 3.1.6.5 | User | |
| Fig 3.1.7.1 | Home Page | 31 |
| Fig 3.1.7.2 | Admin login | |
| Fig 3.1.7.3 | Publisher Registration | 32 |
| Fig 3.1.7.4 | Publisher login | |
| Fig 3.1.7.5 | Publisher Registration list | 33 |
| Fig 3.1.7.6 | Create Event and Competition | |
| Fig 3.1.7.7 | Posted Events | 34 |
| Fig 3.1.7.8 | Manage and view events | |
| Fig 3.1.7.9 | User Event list | 35 |

| | | |
|--------------|--------------------------------|----|
| Fig 3.1.7.10 | Apply for events | |
| Fig 3.1.7.11 | View Applied participants list | 36 |
| Fig 3.1.8.1 | System Architecture | 43 |