

EVENTOR EVENT PLANNER

Event Management System

Project Report Submitted by

Sarath Gopalakrishnan

Reg. No: LAJC18MCA052

In Partial fulfillment for the award of the degree

Of

MASTER OF COMPUTER APPLICATIONS (MCA)

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY



**AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY**

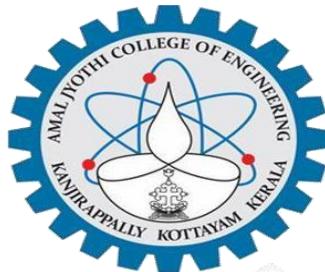
[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala - 686518]

2019-2021

AMAL JYOTHI COLLEGE OF ENGINEERING

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE,
Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala - 686518]

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS



CERTIFICATE

This is to certify that the project entitled "**Eventor Event Planner – Event Management System**" is a bonafide record of the work done by **Sarath Gopalakrishnan LAJC18MCA052**, during the academic year **2019-2021** carried out under our supervision. It is certified that all corrections/suggestions indicated for assessment have been incorporated in the report. This work report has been approved as it satisfies the academic requirements in respect of the project work prescribed by the university for the Master of Computer Applications Degree. Certified further, that to the best of our knowledge the exact work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this to any other candidate.

Fr. Dr. Rubin Thottupurathu Jose Sr. Elsin Chakkalackal Ms. Gloriya Mathew

Head of the Department

Project Coordinator

Project Supervisor

External Expert appointed by the university

DECLARATION

I hereby declare that the project report “**Eventor Event Planner – Event Management System**” is a bonafide work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Degree of Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2019-2021.

Date.....

Sarath Gopalakrishnan

KANJIRAPPALLY

Reg. No: LAJC18MCA052

ACKNOWLEDGEMENT

First and foremost, I thank Almighty God for his gracious guidance through the project. I take this opportunity to express my gratitude to all those who have helped me in completing the project successfully.

It has been said that gratitude is the memory of the heart. I acknowledge my deep sense of gratitude to our manager **Rev. Fr. Dr. Mathew Paikatt** for providing all the infrastructural facilities for us, our Principal **Dr. Z V Lakaparampil** for providing good faculty for guidance.

I take the immense pleasure in expressing my thanks to Head of the Department of Master of Computer Applications, **Rev. Fr. Dr. Rubin Thottupurathu Jose**, for his kind patronages in making this project a successful one. I would like to extend my sincere thanks to our project coordinator **Sr. Elsin Chakkalackal** and my project guide **Ms. Gloriya Mathew** for her guidance and cooperation, without which this would not have been a success.

I am indebted to my beloved teachers whose cooperation and suggestions throughout the project which helped me a lot. I also thank all my friends and classmates for their interest, dedication and encouragement shown towards the project. I convey hearty thanks to parents for the moral support, suggestion and encouragement to make this venture a success.

Sarath Gopalakrishnan

ABSTRACT

“EVENTOR EVENT PLANNER” is a web application project. This project is done to make the manual process to computerized. This System is an event organizing website which brings the facility for the users to book all the events related with a program such as wedding, birth day party or college events etc. The client has to visit the service providers to book all the events or have to visit different offices to book the events for the same program. This reduces the complexity of the users.

The cost of different types of events can be seen through this website. This website includes all event related functions such as food, venue, photo, video etc. The users can view different companies and their foods, decorations, photo, video etc and their cost also. This website reduces the complexities related with the program arrangements and reduces the workload of the user.

CONTENT

Sl. No	Topic	Page No
	Introduction	1
1	Using Git as a Version Control System	2
1.1	Introduction to GitHub	3
1.2	Working With Git	4-6
2	Project Documentation	7
2.1	Introduction	8
2.1.1	Project Overview	8
2.1.2	Project Specification	8
2.2	System Study	9
2.2.1	Introduction	9
2.2.2	Existing system	10
2.2.3	Proposed system	10-11
2.3	Requirement Analysis	11
2.3.1	Feasibility Study	11
2.3.1.1	Economic Feasibility	12
2.3.1.2	Technical Feasibility	12
2.3.1.3	Behavioral Feasibility	13
2.4	Requirement Modeling	13
2.4.1	UML Diagrams	13-25
2.5	System Specification	26
2.5.1	Hardware Specification	26
2.5.2	Software Specification	26
2.6	Software Description	26
2.6.1	PHP	26
2.6.2	MySQL	27-28

2.7	System Design	28
2.7.1	Architectural Design	29
2.7.2	Module Design	29-31
2.7.3	Data Base Design	31-46
2.8	System Testing	46
2.8.1	Introduction	46
2.8.2	Test Plan	47
2.8.2.1	Unit Testing	48
2.8.2.2	Integration Testing	49
2.8.2.3	Validation Testing	49
2.8.2.4	User Acceptance Testing	49
2.8.3	Test Case	50-51
2.9	Implementation	51
2.9.1	Implementation Procedure	52
2.9.2	User Training	53
2.9.3	Operational Document	53
2.9.4	System Maintenance	53
2.10	Conclusion & Future Enhancements	54
2.10.1	Future Enhancement	54
2.10.2	Conclusion	54
2.11	Bibliography	55
2.12	Appendix	56
2.12.1	Sample Code	56-94
2.12.2	Screen Shots	95-104

LIST OF ABBREVIATIONS

IDE -	Integrated Development Environment
HTML -	Hyper Text Markup Language
CSS -	Cascading Style Sheet
SQL -	Structured Query Language
UML -	Unified Modeling Language

INTRODUCTION

“EVENTOR EVENT PLANNER” is a web application project. This project is done to make the manual process to computerized. This System is an event organizing website which brings the facility for the users to book all the events related with a program such as wedding, birth day party or college events etc. The client has to visit the service providers to book all the events or have to visit different offices to book the events for the\ same program. This reduces the complexity of the users. The cost of different types of events can be seen through this website. This website includes all event related functions such as food, venue, photo, video etc. The users can view different companies and their foods, decorations, photo, video etc and their cost also. This website reduces the complexities related with the program arrangements and reduces the workload of the user.

The proposed system is a website in which user can buy book events. The main objective of the ‘EVENTOR EVENT PLANNER’ is to manage the details of the companies and users. The overall control is done by the administrator and only administrator and companies can guarantee the access. The main purpose of the project is to reduce the manual work.

1. USING GIT AS A VERSION CONTROL SYSTEM

1.1 Introduction to GitHub

GitHub is a web-based version-control and collaboration platform for software developers. GitHub, which is delivered through a software-as-a-service (SaaS) business model, was started in 2008 and was founded on Git, an open source code management system created by Linus Torvalds to make software builds faster. And it is used to store the source code for a project and track the complete history of all changes to that code. It allows developers to collaborate on a project more effectively by providing tools for managing possibly conflicting changes from multiple developers. GitHub allows developers to change, adapt and improve software from its public repositories for free, but it charges for private repositories, offering various paid plans. Each public or private repository contains all of a project's files, as well as each file's revision history. Repositories can have multiple collaborators and can be either public or private.

GitHub facilitates social coding by providing a web interface to the Git code repository and management tools for collaboration. GitHub can be thought of as a serious social networking site for software developers. Members can follow each other, rate each other's work, receive updates for specific projects and communicate publicly or privately.

GitHub products and features

GitHub offers an on-premises version in addition to the well-known SaaS product. GitHub Enterprise supports integrated development environments and continuous integration tool integration, as well as a litany of third-party apps and services. It offers increased security and auditability than the SaaS version.

1.2 Working with Git

- Click on sign up button and create an account in GitHub.com after that sign in to the Github.



Sign in to GitHub

Username or email address

Password [Forgot password?](#)

[Sign in](#)

New to GitHub? [Create an account.](#)

[Terms](#) [Privacy](#) [Security](#) [Contact GitHub](#)

□ Create a repository in GitHub

The screenshot shows the GitHub interface for creating a new repository. At the top, there's a search bar and navigation links for Pull requests, Issues, Marketplace, and Explore. Below that, the main title is 'Create a new repository'. A sub-instruction says 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.' Under 'Repository template', there's a dropdown menu set to 'No template'. The 'Owner' field shows 'SarahGopalakrishnan24' with a dropdown arrow, and the 'Repository name' field is filled with 'EVENTOR'. A note below says 'Great repository names are short and memorable. Your new repository will be created at [EVENTOR.github-spoon](#)'. There's a 'Description (optional)' input field, a 'Public' radio button (selected), and a 'Private' radio button. Below these, under 'Initialize this repository with:', there are four options: 'Add a README file' (unchecked), 'Add a .gitignore' (unchecked), 'Choose a license' (unchecked), and 'Skip this step if you're importing an existing repository'. At the bottom right is a green 'Create repository' button.

□ Once repository is completed, you can setup the repository.

The screenshot shows the GitHub repository page for 'Eventor-Event-Planner'. The header includes a search bar, navigation links for Pull requests, Issues, Marketplace, and Explore, and user stats (Unwatch 1, Star 0, Fork 0). Below the header, the repository name 'SarahGopalakrishnan24 / Eventor-Event-Planner' is shown. The main content area has a heading 'Quick setup — if you've done this kind of thing before' with instructions for setting up on Desktop or via HTTPS/SSH, and a note about creating a new file or uploading an existing one. It also mentions initializing every repository with a README, LICENSE, and .gitignore. Below this, there are sections for '...or create a new repository on the command line' (with a code snippet) and '...or push an existing repository from the command line' (with another code snippet). At the bottom, there's a section for '...or import code from another repository' with a note about initializing the repository with code from Subversion, Mercurial, or TFS projects, and a 'Import code' button. A 'ProTip!' note at the bottom right suggests using the URL for adding GitHub as a remote.

- Download and install Github desktop application. Once installed Go to Repository>Clone Repository, and select your repository which was created in GitHub.com or select a local system folder.
- Push the local repository to origin.
- Refresh GitHub.com repository to fetch commits.

SarathGopalakrishnan24 / EVENTOR

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master · 1 branch · 0 tags

Go to file Add file Code

SarathGopalakrishnan24 Add files via upload 7e47711 14 days ago 9 commits

EVENT MANAGEMENT DOCUMENTAT... Add files via upload 10 months ago

EVENT MANAGEMENT NEW FORM D... Add files via upload 11 months ago

EVENT MANAGEMENT TABLE DESIGN... Add files via upload 11 months ago

EVENT MANAGEMENT FORM DESIGN... Add files via upload 10 months ago

EVENT MANAGEMENT FORM DESIGN... Add files via upload 12 months ago

EVENT MANAGEMENT SYSTEM.pdf Add files via upload 12 months ago

EVENT MANAGEMENT TABLE DESIGN... Add files via upload 10 months ago

EVENT MANAGEMENT UML DIAGRA... Add files via upload 11 months ago

EVENT MANGEMENT NEW TABLE DE... Add files via upload 11 months ago

EVENT MANGEMENT TABLE DESIGN... Add files via upload 12 months ago

EVENTOR.FORM DESIGN.pdf Add files via upload 3 months ago

EVENTOR.TABLE DESIGN.pdf Add files via upload 3 months ago

README.md Initial commit 12 months ago

RESEARCH PAPER ABSTRACT.pdf Add files via upload 3 months ago

SarathGopalakrishnan24(Project Report)... Add files via upload 14 days ago

UML DIAGRAMS OF EVENT MANAGE... Add files via upload 10 months ago

UML DIAGRAMS OF EVENTOR.pdf Add files via upload 3 months ago

About No description, website, or topics provided.

Readme

Releases No releases published Create a new release

Packages No packages published Publish your first package

Contributors 2

SarathGopalakrishnan24 Sarath Gop... SarathGopalakrishnan24 Sarath Gopal...

README.md

EVENTOR

2. PROJECT DOCUMENTATION

2.1 INTRODUCTION

2.1.1 Project Overview

“EVENTOR EVENT PLANNER” is a web application project. This project is done to make the manual process to computerized. This System is an event organizing website which brings the facility for the users to book all the events related with a program such as wedding, birth day party or college events etc. The client has to visit the service providers to book all the events or have to visit different offices to book the events for the same program.

This reduces the complexity of the users. The cost of different types of events can be seen through this website. This website includes all event related functions such as food, venue, photo, video etc. The users can view different companies and their foods, decorations, photo, video etc and their cost also. This website reduces the complexities related with the program arrangements and reduces the workload of the user.

2.1.2 Project Specification

The proposed system is a website in which user can buy book events. The main objective of the ‘EVENTOR EVENT PLANNER’ is to manage the details of the companies and users. The overall control is done by the administrator and only administrator and companies can guarantee the access. The main purpose of the project is to reduce the manual work.

The system includes 3 modules: They are:

1. Admin Module

Admin must have a login into this system. He has the overall control of the system. Admin can add or update event categories etc.. Admin can View all the registered companies, users and can able to approve or reject companies and users. Admin is an authorized person and he is in full control of the overall system..

2. Company Module

The Company has the role to take different event management works.

3. User Module

The user is the person who can View & Book an event related functionalities that organized by companies.

2.2 SYSTEM STUDY

2.2.1 Introduction

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers. System analysis or study is an important phase of any system development process. The system is studied to the minute's detail and analyzed. The system analyst plays the role of the interrogator and dwells deep into the working of the present system. The system is viewed as a whole and the input to the system are identified. The outputs from the organizations are traced to the various processes. System analysis is concerned with becoming aware of the problem, identifying the relevant and decisional variables, analyzing and synthesizing the various factors and determining an optimal or at least a satisfactory solution or program of action.

A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to a conclusion. The conclusion is an understanding of how the system functions. This system is called the existing system. Now the existing system is subjected to close study and problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

Preliminary study is the process of gathering and interpreting facts, using the information for further studies on the system. Preliminary study is problem solving activity that requires intensive communication between the system users and system developers. It does various feasibility studies. In these studies, a rough figure of the system activities can be obtained, from which the decision about the strategies to be followed for effective system study and analysis can be taken.

2.2.2 EXISTING SYSTEM

The current system is a manual one where most of the processes are done without using the computer. The person has to visit a particular office for an enquiry. The existing system's process is quite complex and manual one. The Event management system has to keep all records of events manually. In existing system there are lot of paper work and manual processing. While writing on a paper the manager has to keep all the records very carefully because entire data are written in those books. Everything is paper based hence it is very time consuming. More than one person cannot access the data at same time. If any customer wants to organize event such as Birthday Party, Marriage, Reception, Ring Ceremony than they have to contact various event organizer and book the package of event manually.

Drawbacks of existing system

- Very time and effort consuming.
- Needs extra cost.
- Lot of paper work required.
- Maintenance is difficult.
- Chances to lost data.
- More than one person cannot access the data on the same time.
- Calculations are manually done so there is a chance for occurring errors.

2.2.3 PROPOSED SYSTEM

The proposed system can overcome all the limitations of the existing system. The system provides proper security by using the unique username and password and reduce the manual work. It is a paperless work, so there is no loss of data. This overall increase management productivity, eliminate paper works, reduce manpower, and prove to be very economical in the long run. This will immediately reduce the manual processing, thereby increasing the speed of event management process. All event related functions can be booked through this website.

Advantages of proposed system

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features:

- Ensure data accuracy: -

The proposed system eliminates the manual errors while entering the details of the users during the registration.

- Better security: -

For data to remain secure measures must be taken to prevent unauthorized access. Security means that data are protected from various forms of destruction. The system security problem can be divided into four related issues: security, integrity, privacy and confidentiality. Username and password requirement to sign in ensures security. It will also provide data security as we are using the secured databases for maintaining the documents.

- Better service: -

The product will avoid the burden of hard copy storage. We can also conserve the time and human resources for doing the same task. The data can be maintained for longer period with no loss of data.

- Greater Efficiency: -.

This system is to accomplish something with the least waste of time and effort; competency in performance.

2.3 REQUIREMENT ANALYSIS

2.3.1 Feasibility study

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study let's the developer foresee the future of the project and the usefulness. A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibilities. The following are its features: -

2.3.1.1 Economic Feasibility

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

The proposed system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development.

The cost of project, EVENTOR EVENT PLANNER was divided according to the system used, its development cost and cost for hosting the project. According to all the calculations the project was developed in a low cost. As it is completely developed using open source software.

2.3.1.2 Technical Feasibility

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures. Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed. Technical issues raised during the investigation are:

- Does the existing technology sufficient for the suggested one?
- Can the system expand if developed?

The project should be developed such that the necessary functions and performance are achieved within the constraints. The project requires High Resolution Scanning device and utilizes Cryptographic techniques. Through the technology may become obsolete after some period of

time, due to the fact that newer version of same software supports older versions, the system may still be used. So there are minimal constraints involved with this project. The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible for development. The system has been developed using PHP in front end and MySQL in server in back end, the project is technically feasible for development. The System used was also of good performance of Processor Intel i3 core; RAM 8GB and, Hard disk 1TB.

2.3.1.3 Behavioral Feasibility

The proposed system includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

EVENTOR EVENT PLANNER, GUI is simple so that users can easily use it. EVENTOR EVENT PLANNER is simple enough so that no training is needed.

2.4 Requirement Modeling

2.4.1 UML diagram

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.

UML stands for Unified Modeling Language. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general-purpose visual modeling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object-oriented analysis and design. After some standardization, UML has become an OMG

standard. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete. UML includes the following nine diagrams.

- Activity diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- State chart diagram
- Class diagram
- Object diagram
- Component diagram
- Deployment diagram

1. UML Activity Diagram

Activity Diagram describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not.

UML ACTIVITY DIAGRAM

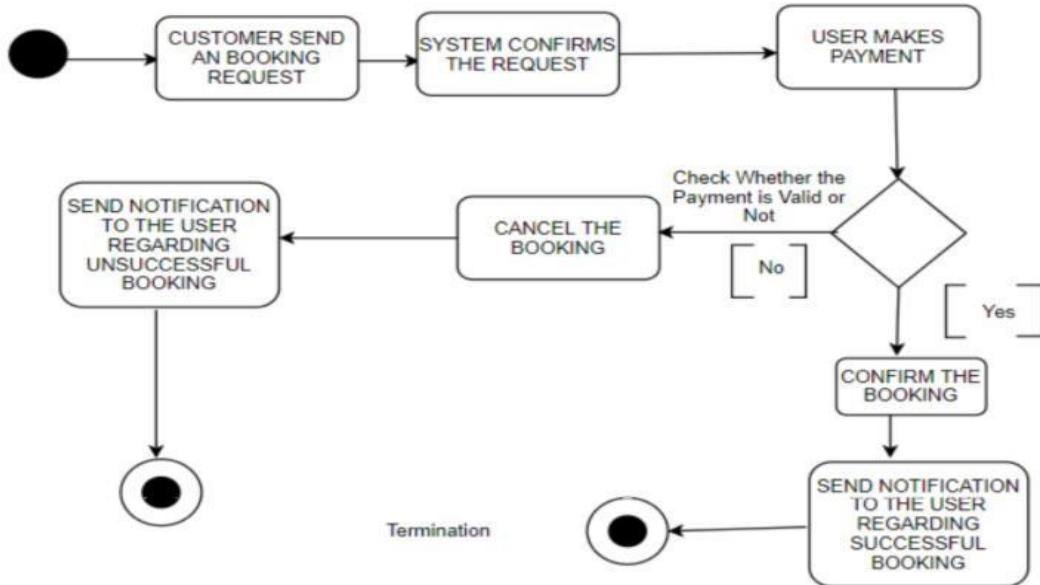


Fig 1: Activity Diagram of Customer Send Order Request

2. UML Use Case Diagram

A use case diagram is a graphic depiction of the interactions among the elements on a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems. System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service-oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating, payment processing, and customer relations. A use case diagram contains four components.

- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to their roles.

- The use cases, which are the specific roles are played by the actors within and around the system.
- The relationships between and among the actors and the use cases.

Use case diagrams are drawn to capture the functional requirements of a system.

After identifying the above items, we have to use the following guidelines to draw an efficient use case diagram.

- The name of a use case is very important. The name should be chosen in such a way so that it can identify the functionalities performed.
- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships, as the main purpose of the diagram is to identify the requirements.
- Use notes whenever required to clarify some important points.

UML USECASE DIAGRAM OF ADMIN

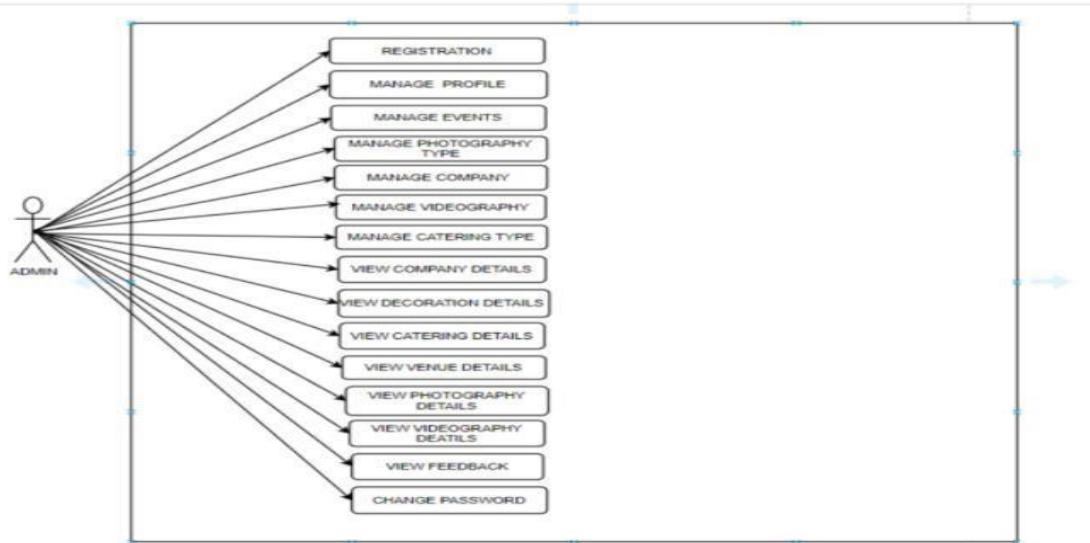


Fig 2: Use Case Diagram of Admin

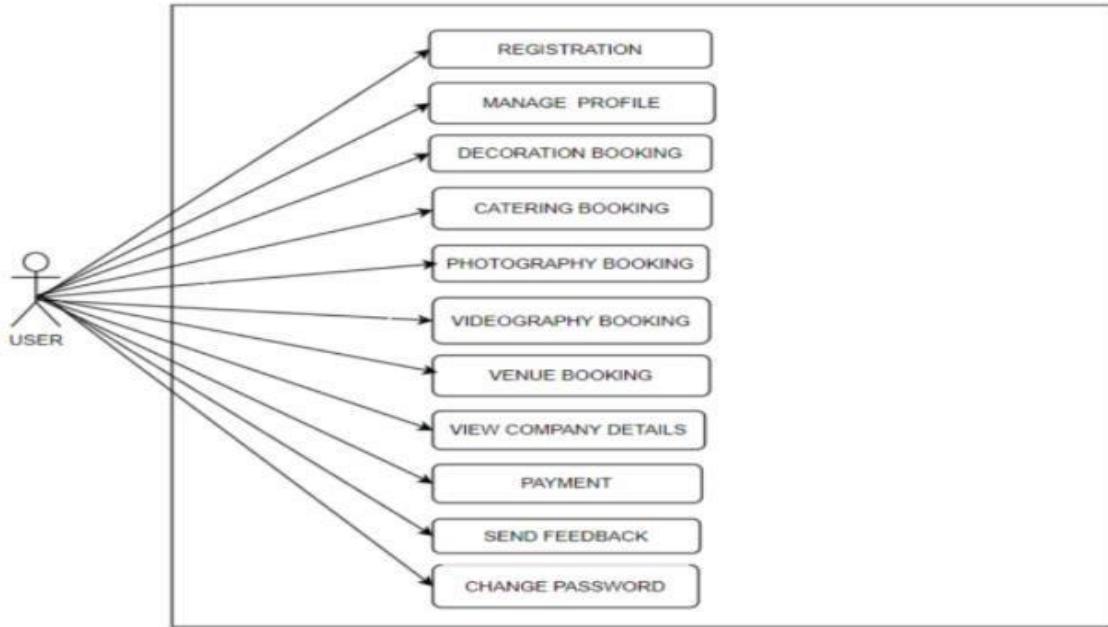
UML USECASE DIAGRAM OF USER

Fig 3: Use Case Diagram of User

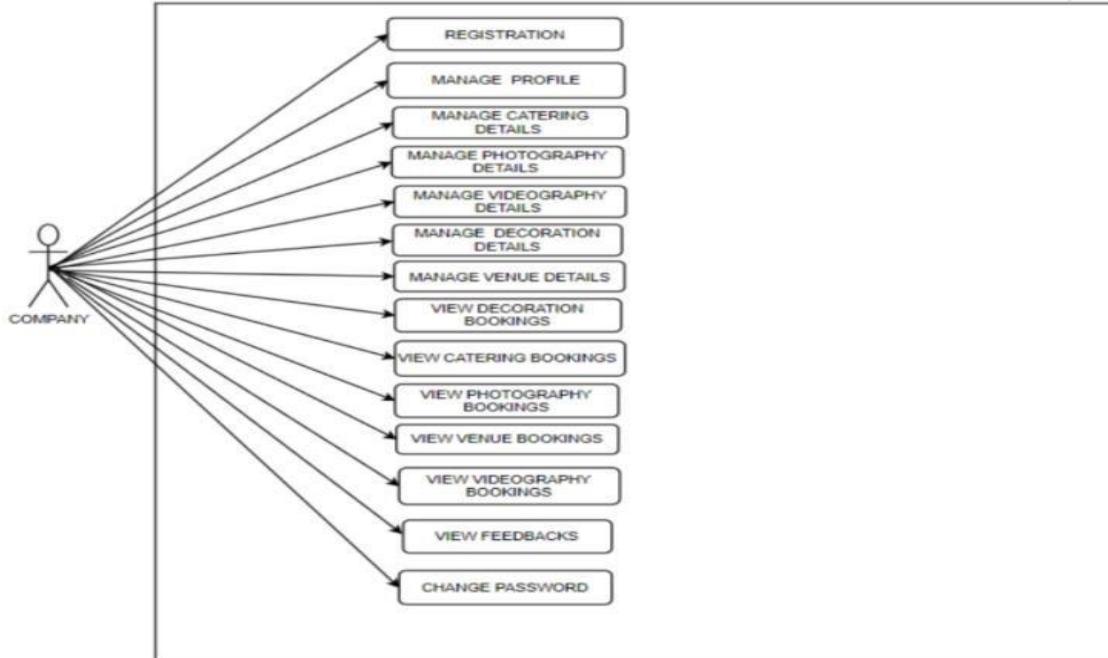
UML USECASE DIAGRAM OF COMPANY

Fig 4: Use Case Diagram of Company

3. UML Sequence Diagram

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart.

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios. A sequence diagram shows, as parallel vertical lines (*lifelines*), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

UML SEQUENCE DIAGRAM

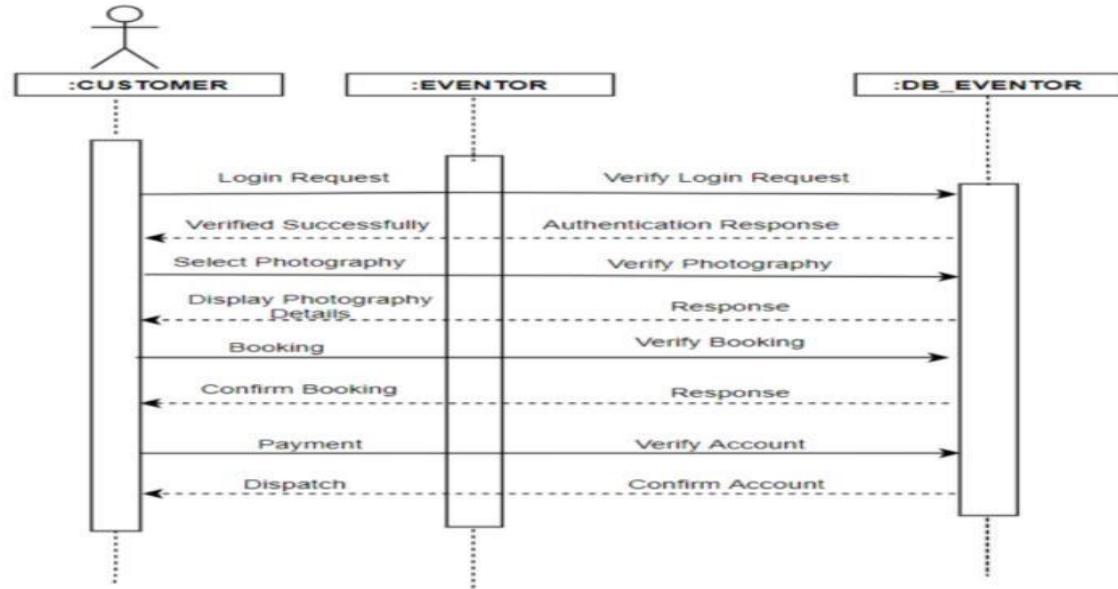


Fig 5: Sequence Diagram of Customer Place Order and Makes Payment

4. UML Collaboration Diagram

The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system.

Notations of a Collaboration Diagram

Following are the components of a component diagram that are enlisted below:

1. **Objects:** The representation of an object is done by an object symbol with its name and class underlined, separated by a colon. In the collaboration diagram, objects are utilized in the following ways:
 - The object is represented by specifying their name and class.
 - It is not mandatory for every class to appear.
 - A class may constitute more than one object.
 - In the collaboration diagram, firstly, the object is created, and then its class is specified.
 - To differentiate one object from another object, it is necessary to name them.
2. **Actors:** In the collaboration diagram, the actor plays the main role as it invokes the interaction. Each actor has its respective role and name. In this, one actor initiates the use case.
3. **Links:** The link is an instance of association, which associates the objects and actors. It portrays a relationship between the objects through which the messages are sent. It is represented by a solid line. The link helps an object to connect with or navigate to another object, such that the message flows are attached to links.

4. Messages: It is a communication between objects which carries information and includes a sequence number, so that the activity may take place. It is represented by a labeled arrow, which is placed near a link. The messages are sent from the sender to the receiver, and the direction must be navigable in that particular direction. The receiver must understand the message.

The collaborations are used when it is essential to depict the relationship between the object. Both the sequence and collaboration diagrams represent the same information, but the way of portraying it quite different. The collaboration diagrams are best suited for analyzing use cases.

UML COLLABORATION DIAGRAM

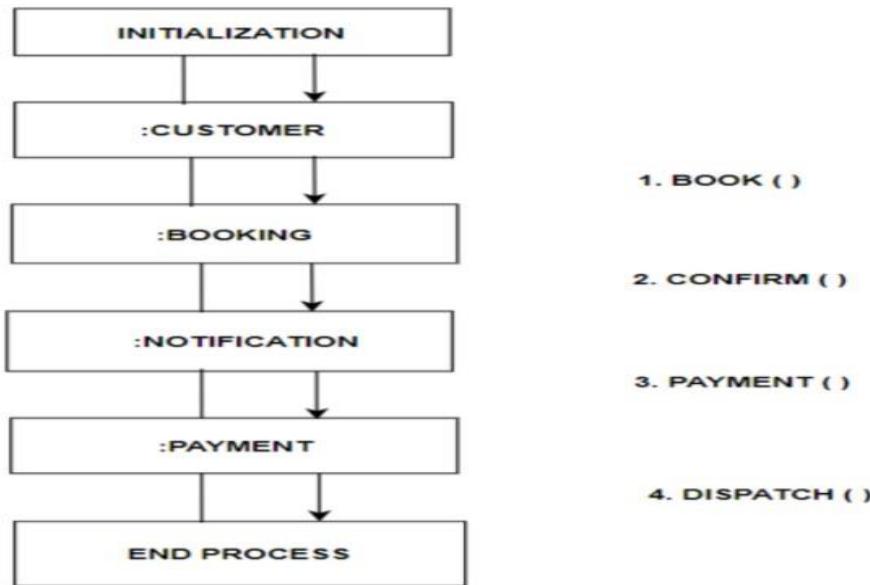


Fig 6: Collaboration Diagram of Customer Search Events and Makes Payment

5. UML State Chart Diagram

It describes different states of a component in a system. The states are specific to a component/object of a system. A Statechart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events. They define different states

of an object during its lifetime and these states are changed by events. Statechart diagrams are useful to model the reactive systems. Reactive systems can be defined as a system that responds to external or internal events.

Statechart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered. The most important purpose of Statechart diagram is to model lifetime of an object from creation to termination. Statechart diagrams are also used for forward and reverse engineering of a system.

However, the main purpose is to model the reactive system.

Following are the main purposes of using Statechart diagrams –

- To model the dynamic aspect of a system.
- To model the life time of a reactive system.
- To describe different states of an object during its life time.
- Define a state machine to model the states of an object.

UML STATECHART DIAGRAM

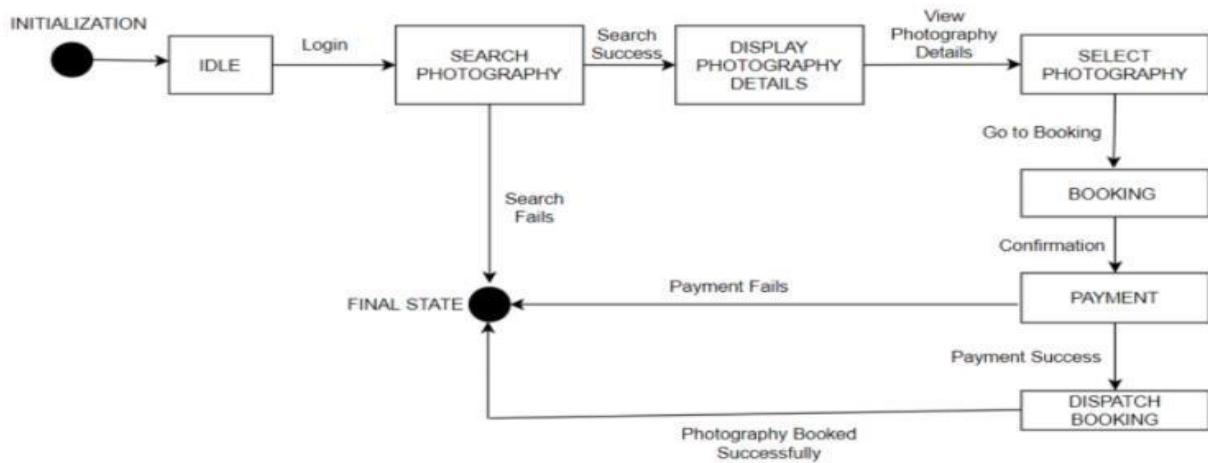


Fig 7: Statechart Diagram of Customer View Events and Makes Payment

6. UML Class Diagram

Class diagram is a static diagram. It represents the static view of an application.

Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

The purpose of the class diagram can be summarized as –

- Analysis and design of the static view of an application.
- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.

UML CLASS DIAGRAM

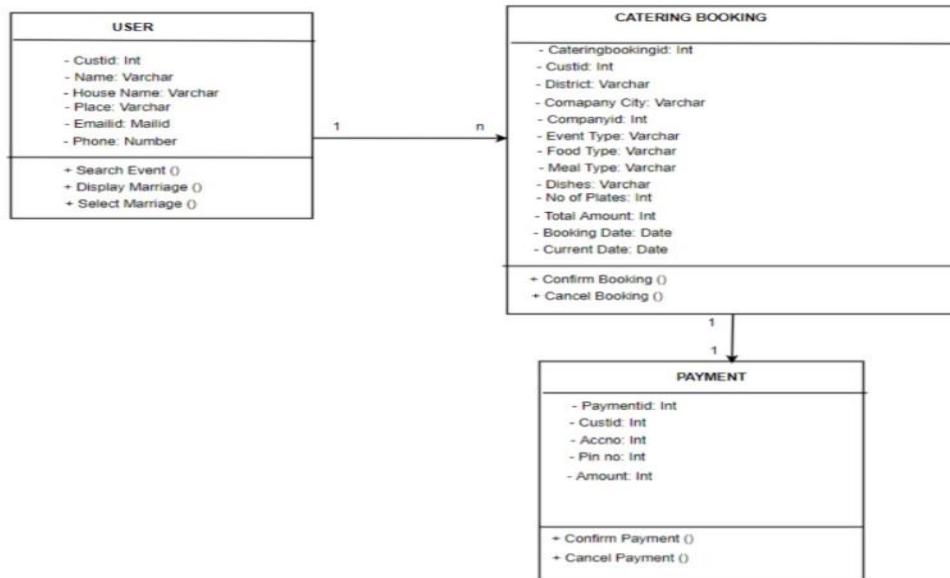


Fig 8: Class Diagram of Customer Add Book Events to Cart

7. Uml Object Diagram

Object diagrams are derived from class diagrams so object diagrams are dependent upon class diagrams. Object diagrams represent an instance of a class diagram. The basic concepts are similar for class diagrams and object diagrams.

Object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moment. Object diagrams are used to render a set of objects and their relationships as an instance.

The purpose of the object diagram can be summarized as –

- Forward and reverse engineering.
- Object relationships of a system.
- Static view of an interaction.
- Understand object behavior and their relationship from practical perspective

UML OBJECT DIAGRAM

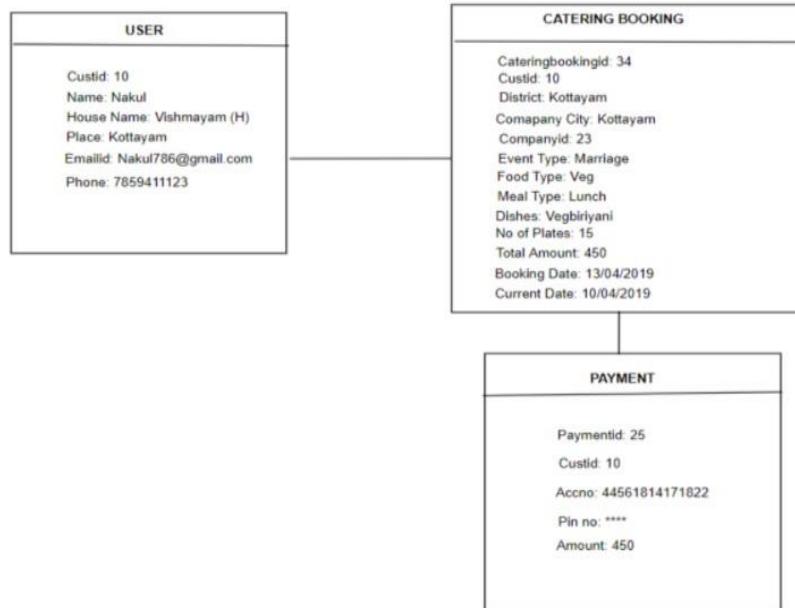


Fig 9: Object Diagram of Customer Add Book Events to Cart

8. Component Diagram

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities. Thus from that point of view, component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files, etc.

Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment. A single component diagram cannot represent the entire system but a collection of diagrams is used to represent the whole.

The purpose of the component diagram can be summarized as –

- Visualize the components of a system.
- Construct executable by using forward and reverse engineering.
- Describe the organization and relationships of the components.

UML COMPONENT DIAGRAM

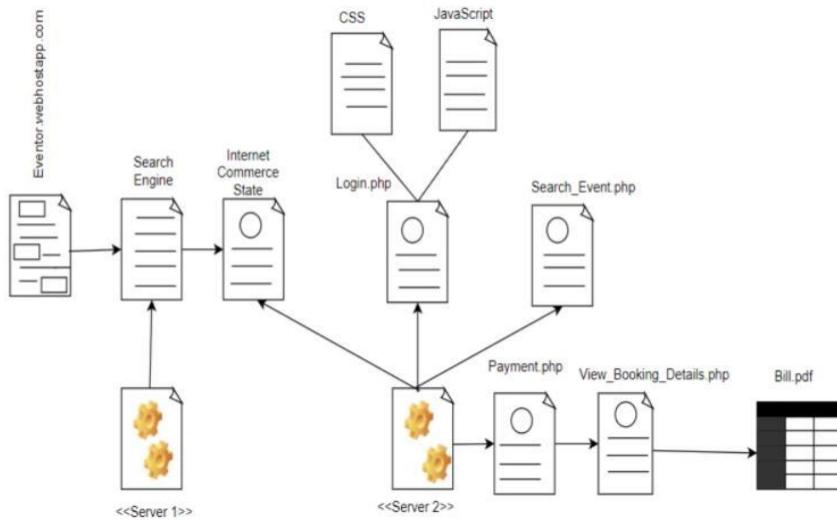


Fig 10: Component Diagram of Customer Login to System and Search for Events

9. Deployment Diagram

Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed. Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

It ascertains how software is deployed on the hardware. It maps the software architecture created in design to the physical system architecture, where the software will be executed as a node. Since it involves many nodes, the relationship is shown by utilizing communication paths.

UML DEPLOYMENT DIAGRAM

Deployment Diagram

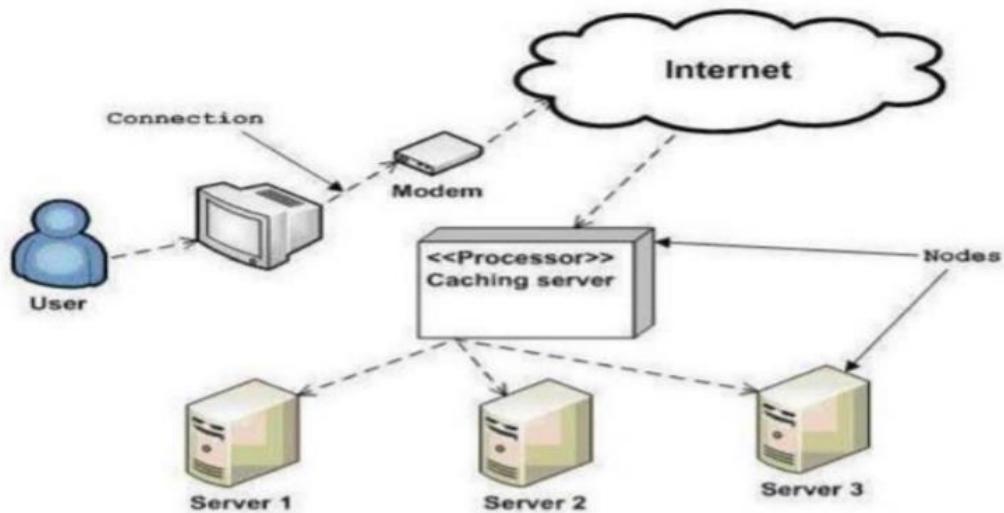


Fig 11: Deployment Diagram of Eventor Event Planner

2.5 System specification

2.5.1 Hardware Specification

Processor	- Intel core i3
RAM	- 8 GB
Hard disk	- 1 TB

2.5.2 Software Specification

Front End	- HTML, CSS
Backend	- PHP
Database	- MYSQL
Client on PC	- Windows 7 and above.
Technologies used	- JS, HTML5, J Query, PHP, CSS, Ajax

2.6 Software Description

2.6.1 PHP

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language. PHP is now installed on more than 244 million websites and 2.1 million web servers. Originally created by Rasmus Ledorf in 1995, the reference implementation of PHP is now produced by the PHP group. While PHP originally stood for personal Home page, it now stands for PHP: Hypertext Preprocessor, a recursive acronym. PHP code is interpreted by a web server with a PHP processor module which generates the resulting web page. PHP commands can be embedded directly into a HTML source document rather than calling an external file to process data. It has also evolved to include a command-line interface capability and can be used in standalone incompatible with the GNU General Public License (GPL) due to restrictions on the usage of the term PHP. PHP can be deployed on most web servers and also as a standalone shell on almost every operating system and platform, free of charge.

2.6.2 MySQL

MySQL, the most popular Open-Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. The MySQL Web site provides the latest information about MySQL software.

- **MySQL is a database management system.**

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

- **MySQL databases are relational.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data. The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax. SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, “SQL92” refers to the standard released in 1992, “SQL: 1999” refers to the standard released in 1999, and “SQL: 2003” refers to the current version of the standard. We use the phrase “the SQL standard” to mean the current version of the SQL Standard at any time.

- **MySQL software is Open Source.**

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information.

- **The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available.

- **MySQL Server works in client/server or embedded systems.**

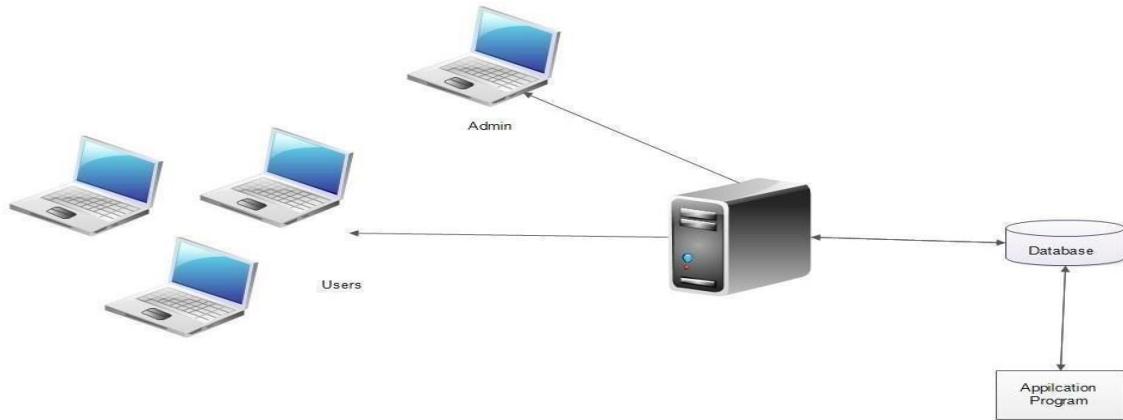
The MySQL Database Software is a client/server system that consists of a multithreaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

2.7 System design

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system

in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user-oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

2.7.1 Architectural Design



The registered customer, seller, and admin can access the website through internet using their Laptop, Smart Phone, Tablet or Desktop Computer. The System's application program processes the user's request and provides the required services by taking data from the system database.

2.7.2 Module Design

Admin Module

The administrator of the company is allowed to access all the services in the system.

Admin has the overall control of the system. Admin can add or update Event categories. Admin can View all the registered Users, Company can able to approve or reject Users and Company.

Manage user details, Add Event Categories	Approve/Reject the registered Company and Users.
View order details	View/Replay to messages and requests

Registered User Module

After registration, user can view events, search events, can add events to cart, purchase event booking through secure online payment.

Customer registration, login	Search events/Add events to cart
Checkout and Order Event Booking	Make payment
View Order Details and download order details	Send complaints and feedbacks

Registered Company Module

The company can register into this web site and once the admin approves the company, they can login and view events. Company can add their events and view their added events. If the events is not available the company can send request to the admin to add their events to the events list.

Company registration, login	View all available events
Add events	View added events
Send Request for adding events	View replays of send requests

2.7.3 Database Design

A database is an organized mechanism that has the capability of storing information through which a user can retrieve stored information in an effective and efficient manner. The data is the purpose of any database and must be protected.

The database design is a two-level process. In the first step, user requirements are gathered together and a database is designed which will meet these requirements as clearly as possible. This step is called Information Level Design and it is taken independent of any individual DBMS.

In the second step, this Information level design is transferred into a design for the specific DBMS that will be used to implement the system in question. This step is called Physical Level Design, concerned with the characteristics of the specific DBMS that will be used. A database design runs parallel with the system design. The organization of the data in the database is aimed to achieve the following two major objectives.

- Data Integrity
- Data independence

Relational Database Management System (RDBMS)

A relational model represents the database as a collection of relations. Each relation resembles a table of values or file of records. In formal relational model terminology, a row is called a tuple, a column header is called an attribute and the table is called a relation. A relational database consists of a collection of tables, each of which is assigned a unique name. A row in a tale represents a set of related values.

Relations, Domains & Attributes

A table is a relation. The rows in a table are called tuples. A tuple is an ordered set of n elements. Columns are referred to as attributes. Relationships have been set between every table in the database. This ensures both Referential and Entity Relationship Integrity. A domain D is a set of atomic values. A common method of specifying a domain is to specify a data type from which the data values forming the domain are drawn. It is also useful to specify a name for the domain to help in interpreting its values. Every value in a relation is atomic, that is not decomposable.

Relationships

- Table relationships are established using Key. The two main keys of prime importance are Primary Key & Foreign Key. Entity Integrity and Referential Integrity Relationships can be established with these keys.
- Entity Integrity enforces that no Primary Key can have null values.
- Referential Integrity enforces that no Primary Key can have null values.
- Referential Integrity for each distinct Foreign Key value, there must exist a matching Primary Key value in the same domain. Other key are Super Key and Candidate Keys.

Normalization

Data are grouped together in the simplest way so that later changes can be made with minimum impact on data structures. Normalization is formal process of data structures in manners that eliminates redundancy and promotes integrity.

Normalization is a technique of separating redundant fields and breaking up a large table into a smaller one. It is also used to avoid insertion, deletion, and updating anomalies.

Normal form in data modelling use two concepts, keys and relationships. A key uniquely identifies a row in a table.

There are two types of keys, primary key and foreign key. A primary key is an element or a combination of elements in a table whose purpose is to identify records from the same table. A foreign key is a column in a table that uniquely identifies record from a different table. All the tables have been normalized up to the third normal form. As the name implies, it denotes putting things in the normal form. The application developer via normalization tries to achieve a sensible organization of data into proper tables and columns and where names can be easily correlated to the data by the user. Normalization eliminates repeating groups at data and thereby avoids data redundancy which proves to be a great burden on the computer resources. These include:

- ✓ Normalize the data.
- ✓ Choose proper names for the tables and columns.
- ✓ Choose the proper name for the data.

First Normal Form

The First Normal Form states that the domain of an attribute must include only atomic values and that the value of any attribute in a tuple must be a single value from the domain of that attribute.

In other words, 1NF disallows “relations within relations” or “relations as attribute values within tuples”. The only attribute values permitted by 1NF are single atomic or indivisible values. The first step is to put the data into First Normal Form. This can be done by moving data into separate tables where the data is of similar type in each table. Each table is given a Primary Key or Foreign Key as per requirement of the project. In this we form new relations for each non- atomic attribute or nested relation. This eliminated repeating groups of data. A relation is said to be in first normal form if only if it satisfies the constraints that contain the primary key only.

Second Normal Form

According to Second Normal Form, for relations where primary key contains multiple attributes, no non-key attribute should be functionally dependent on a part of the primary key. In this we decompose and setup a new relation for each partial key with its dependent attributes. Make sure to keep a relation with the original primary key and any attributes that are fully functionally dependent on it. This step helps in taking out data that is only dependent on a part of the key.

A relation is said to be in second normal form if and only if it satisfies all the first normal form conditions for the primary key and every non-primary key attributes of the relation is fully dependent on its primary key alone.

Third Normal Form

According to Third Normal Form, Relation should not have a non-key attribute functionally determined by another non-key attribute or by a set of non-key attributes. That is, there should be no transitive dependency on the primary key. In this we decompose and set up relation that includes the non-key attributes that functionally determines other non-key attributes. This step is taken to get rid of anything that does not depend entirely on the Primary Key. A relation is said to be in third normal form if only if it is in second normal form and more over the non key attributes of the relation should not be depend on other non-key attribute.

TABLE DESIGN

Table No : 01

Table Name : tbl_login

Primary Key :lid

Table Description: To Store Login Information.

Fieldname	Datatype	Size	Constraint	Description
lid	Int	10	Primary Key	Login id
email	Varchar	30		Mailid
password	Varchar	16		Password
status	Int	2		Status
role	Varchar	20		Role of user

Table No : 02

Table Name : tbl_district

Primary Key :did

Table Description: To Store District Information.

Field Name	Data Type	Size	Constraint	Description
did	Int	2	Primary Key	District Id
dname	varchar	30		District Name
status	Int	2		Active/inactive

Table No : 03

Table Name : tbl_event

Primary Key :eventid

Table Description: To Store Event Category Information.

Field Name	Data Type	Size	Constraint	Description
eventid	Int	20	Primary Key	Event Id
eventtype	Varchar	50		Event Type
status	Int	2		Active/Inactive

Table No : 04**Table Name : tbl_userreg****Primary Key : userid****Foreign Key : did,lid****Table Description: To Store User Registration Information.**

Field Name	Data Type	Size	Constraint	Description
userid	Int	10	Primary Key	User Id
name	Varchar	30	not null	Name of user
address1	Varchar	30	not null	HouseName/Apartment No
address 2	Varchar	20	Not null	City
address 3	Varchar	30	Not null	Post Office
pincode	Int	6	Not null	Pincode
did	Int	2	Foreign Key	District Id referenced from tbl_district
gender	Varchar	10		Gender of the user
dob	date	3		Date of Birth
phone	Number	10		Phone Number
email	email	30		Mail Id
password	password	16		Password
lid	Int	10	Foreign Key	Login id referenced from tbl_login
status	Int	2		Active/Inactive

Table No : 05**Table Name : tbl_companyreg****Primary Key : companyid****Foreign Key : did,lid****Table Description: To Store Company Registration Information.**

Field Name	Data Type	Size	Constraint	Description
companyid	Int	10	Primary Key	Company Id
companynname	Varchar	50		Company Name
address1	Varchar	20		Officename/Building
address2	Varchar	20		City Name
address3	Varchar	20		Post Office
address4	Int	6		PinCode
did	Int	2	Foreign Key	District Id District Id referenced from tbl_district
phoneno	Number	10		Phone Number
emailid	Varchar	50		Email Id
lid	Int	10		Login Id referenced from tbl_login
status	Int	2		Active/Inactive

Table No : 06**Table Name : tbl_cateringcategory****Primary Key : cateringtypeid****Table Description: To Store Catering Category Information.**

Field Name	Data Type	Size	Constraint	Description
cateringtypeid	Int	10	Primary Key	Catering Type Id
foodtype	Varchar	50		Food Type
mealtype	Varchar	50		Mealtype
status	Int	2		Active/Inactive

Table No : 07**Table Name : tbl_cateringdetails****Primary Key : cateringid****Foreign Key : companyid,cateringtypeid****Table Description: To Store Catering Details.**

Field Name	Data Type	Size	Constraint	Description
cateringid	Int	10	Primary Key	Catering Id
companyid	Int	10	Foreign Key	Company Id referenced from tbl_companyreg
cateringtypeid	Int	10	Foreign Key	Catering type id referenced from tbl_cateringitems
dishes	Varchar	50		Type of dishes
amount	Int	7		Amount
status	Int	2		Active/Inactive

Table No : 08**Table Name : tbl_feedbacks****Primary Key : fid****Foreign Key : lid****Table Description: To Store Feedbacks Details.**

Field Name	Data Type	Size	Constraint	Description
fid	Int	10	Primary Key	Feedback Id
feedback	varchar	50		Users Feedback
lid	Int	10	Foreign Key	Login Id referenced from tbl_login
status	Int	2		Active/Inactive

Table No : 09

Table Name : tbl_photographycategory

Primary Key :photographycategoryId

Table Description: To Store Photography Category Information.

Field Name	Data Type	Size	Constraint	Description
photographycategoryId	Int	10	Primary Key	Photographycategory Id
photographycategory	Varchar	10		Type of Photography
status	Int	2		Active/Inactive

Table No : 10

Table Name : tbl_photographydetailes

Primary Key : photographyid

Foreign Key : photographycategoryId,companyid

Table Description: To Store Photography Details.

Field Name	Data Type	Size	Constraint	Description
photographyid	Int	10	Primary Key	Photography Id
companyid	Int	10	Foreign Key	Company Id refrenced from tbl_companyreg
photographycategoryId	Int	10	Foreign Key	Photographycategory Id refrenced from tbl_photographycategory
description	Varchar	50		Description about the Photographycategory
amount	Int	7		Photography Amount
status	Int	2		Active/Inactive

Table No : 11

Table Name : tbl_videographycategory

Primary Key :videographytypeid

Foreign Key :

Table Description: To Store Videography Category Information.

Field Name	Data Type	Size	Constraint	Description
videographytypeid	Int	10	Primary Key	Videographytype Id
videographytype	Varchar	10		Type of Videography
status	Int	2		Active/Inactive

Table No : 12

Table Name : tbl_videographydetails

Primary Key : videographyid

Foreign Key : videographytypeid,companyid

Table Description: To Store Videography Details.

Field Name	Data Type	Size	Constraint	Description
videographyid	Int	10	Primary Key	Videography Id
companyid	Int	10	Foreign Key	Company Id refrenced from tbl_companyreg
videographytypeid	Int	10	Foreign Key	Videographytype Id
description	Varchar	50		Description about the videographytype
amount	Int	7		Videography Amount
status	Int	2		Active/Inactive

Table No : 13

Table Name : tbl_venuedetails

Primary Key : venueid

Foreign Key : companyid

Table Description: To Store Venue Details.

Field Name	Data Type	Size	Constraint	Description
venueid	Int	10	Primary Key	Venue Id
companyid	Int	10	Foreign Key	Company Id referenced from tbl_companyreg
venueplace	Varchar	50		Venue Place
facility	Varchar	10		Ac or Non Ac
image	Varchar	Max		Decoration Image
description	Varchar	50		Description about the image
amount	Int	7		VenueAmount
status	Int	2		Active/Inactive

Table No : 14

Table Name : tbl_decorationdetails

Primary Key : decorationid

Foreign Key : companyid,eventid

Table Description: To Store Decoration Details.

Field Name	Data Type	Size	Constraint	Description
decorationid	Int	10	Primary Key	Decoration Id
companyid	Int	10	Foreign Key	Company Id referenced from tbl_companyreg
eventid	Int	10	Foreign Key	Event Id referenced from tbl_event
image	Varchar	Max		Decoration Image
description	Varchar	50		Description about the image
amount	Int	7		Decoration Amount
Status	Int	2		Active/Inactive

Table No: 15**Table Name: tbl_bookingphotographydetails****Primary Key: bookingphotoid****Foreign Key : userid,companyid,eventid,photographyid****Table Description: To Store Booking Photography Details.**

Field Name	Data Type	Size	Constraint	Description
bookingphotoid	Int	10	Primary Key	Booking Photo Id
userid	Int	10	Foreign Key	User Id referenced from tbl_userreg
companyid	Int	10	Foreign Key	Company Id referenced from tbl_companyreg
eventid	Int	10	Foreign Key	Event Id referenced from tbl_event
noofalbums	Int	2		No of Albums
bookingdate	Date	5		Booking Date
bookingtime	TimeStamp	5		Booking Time
photographyid	Int	10	Foreign Key	Photography Id referenced from tbl_photographydetails
totalamount	Int	6		Total Amount
date	Date	5		Current Date
status	Int	2		Active/Inactive

Table No : 16**Table Name : tbl_bookingvideographydetails****Primary Key : bookingvideoid****Foreign Key : userid,companyid,eventid,videographyid****Table Description: To Store Booking Videography Details.**

Field Name	Data Type	Size	Constraint	Description
bookingvideoid	Int	10	Primary Key	Booking Video Id
userid	Int	10	Foreign Key	User Id referenced from tbl_userreg
companyid	Int	10	Foreign Key	Company Id referenced from tbl_companyreg
eventid	Int	10	Foreign Key	Event Id referenced from tbl_event
bookingdate	Date	5		Booking Date
bookingtime	TimeStamp	5		Booking Time
videographyid	Int	10	Foreign Key	Videography Id referenced from tbl_videographydetails
totalamount	Int	7		Total Amount
date	Date	5		Current Date
status	Int	2		Active/Inactive

Table No : 17

Table Name : tbl_bookingdecorationdetails

Primary Key : bookingdecorationid

Foreign Key : userid,companyid,eventid,decorationid

Table Description: To Store Booking Decoration Details.

Field Name	Data Type	Size	Constraint	Description
bookingdecorationid	Int	10	Primary Key	Booking Decoration Id
userid	Int	10	Foreign Key	User Id referenced from tbl_userreg
companyid	Int	10	Foreign Key	Company Id referenced from tbl_companyreg
eventid	Int	10	Foreign Key	Event Id referenced from tbl_event
bookingdate	Date	5		Booking Date
bookingtime	TimeStamp	5		Booking Time
decorationid	Int	10	Foreign Key	Decoration Id
totalamount	Int	7		Total Amount
date	Date	5		Current Date
status	Int	2		Active/Inactive

Table No : 18

Table Name : tbl_bookingvenuedetails

Primary Key : bookingvenueid

Foreign Key : userid,companyid,eventid,venueid

Table Description: To Store Booking Venue Details.

Field Name	Data Type	Size	Constraint	Description
bookingvenueid	Int	10	Primary Key	Booking Venue Id
userid	Int	10	Foreign Key	User Id referenced from tbl_userreg
companyid	Int	10	Foreign Key	Company Id referenced from tbl_companyreg
venueid	Int	10	Foreign Key	Venue Id from tbl_venue details
eventid	Int	10	Foreign Key	Event Id referenced from tbl_event
no of guests	Int	5		Total no of guests
image	Varchar	MAX		Description about the auditorium

amount	Int	7		Total Venue Amount
bookingdate	Date	5		Booking Date
bookingtime	TimeStamp	5		Booking Time
date	Date	5		Current Date
status	Int	2		Active/Inactive

Table No : 19

Table Name : tbl_bookingcateringdetails

Primary Key : bookingfoodid

Foreign Key : userid,companyid,eventid,cateringid

Table Description: To Store Booking Catering Details.

Field Name	Data Type	Size	Constraint	Description
bookfoodid	Int	10	Primary Key	Booking Food Id
userid	Int	10	Foreign Key	User Id referenced from tbl_userreg
companyid	Int	10	Foreign Key	Company Id referenced from tbl_companyreg
eventid	Int	10	Foreign Key	Event Id referenced from tbl_event
cateringid	Int	10	Foreign Key	Catering Id referenced from tbl_cateringdetails
noofplates	Int	5		No of plates needed
totalamount	Big Int	7		Total Food Cost
bookingdate	Date	5		Booking Date
bookingtime	TimeStamp	5		Booking Time
date	Date	5		Current Date
status	Int	2		Active/Inactive

Table No	: 20
Table Name	: tbl_payment
Primary Key	: paymentid
Foreign Key	: userid
Table Description	: To Store Payment Details.

Field Name	Data Type	Size	Constraint	Description
paymentid	Int	10	Primary Key	Payment Id
userid	Int	10	Foreign key	User id referenced from tbl_userregistration
accno	Int	14		Account No of user
cvv	Int	3		Card verification Value
amount	Int	5		Bank Balance
status	Int	2		Active/inactive

Table No	: 21
Table Name	: tbl_transaction
Primary Key	: tid
Foreign Key	: userid
Table Description	: To Store Transaction Details.

Field Name	Data Type	Size	Constraint	Description
tid	Int	10	Primary Key	Transaction Id
fromacc	Int	14		From Account
toacc	Int	14		To Account
amount	Int	5		Amount
date	Date	3		Date of Transcation
userid	int	10	Foreign Key	User Id referenced from tbl_userregsitartion
status	int	2		Active/inactive

Table No : 22

Table Name :tbl_accountdetails

Primary Key :accounted

Foreign Key :t_id

Table Description: To Store Account Details.

Field Name	Data Type	Size	Constraint	Description
accountid	Int	10	Foreign Key	Account Id referenced from tbl_accountdetails
accountno	Int	14		Account number of the user
pinno	Int	4		Pin no of the user
amount	Int	8		Amount in account
t_id	Int	10	Foreign Key	Transaction Id referenced from tbl_transaction
status	Int	2		Active/Inactive

2.8 System testing

2.8.1 Introduction

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

Other activities which are often associated with software testing are static analysis and dynamic analysis. Static analysis investigates the source code of software, looking for problems and gathering metrics without actually executing the code. Dynamic analysis

looks at the behavior of software while it is executing, to provide information such as execution traces, timing profiles, and test coverage information.

Testing is a set of activity that can be planned in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers based system. Nothing is complete without testing, as it vital success of the system testing objectives, there are several rules that can serve as testing objectives. They are:

Testing is a process of executing a program with the intent of finding an error.

- A good test case is one that has high possibility of finding an undiscovered error.
- A successful test is one that uncovers an undiscovered error.

If a testing is conducted successfully according to the objectives as stated above, it would uncover errors in the software. Also testing demonstrate that the software function appear to be working according to the specification, that performance requirement appear to have been met.

There are three ways to test program.

- For correctness
- For implementation efficiency
- For computational complexity

Test for correctness are supposed to verify that a program does exactly what it was designed to do. This is much more difficult than it may at first appear, especially for large programs.

2.8.2 Test Plan

A test plan implies a series of desired course of action to be followed in accomplishing various testing methods. The Test Plan acts as a blue print for the action that is to be followed. The software engineers create a computer program, its documentation and related data structures. The software developers is always responsible for testing the individual units of the programs, ensuring that each performs the function for which it was designed. There is an independent test group (ITG) which is to remove the inherent

problems associated with letting the builder to test the thing that has been built. The specific objectives of testing should be stated in measurable terms. So that the mean time to failure, the cost to find and fix the defects, remaining defect density or frequency of occurrence and test work-hours per regression test all should be stated within the test plan.

The levels of testing include:

- Unit testing
- Integration Testing
- Data validation Testing
- Output Testing

2.8.2.1 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design – the software component or module. Using the component level design description as a guide, important control paths are tested to uncover errors within the boundary of the module. The relative complexity of tests and uncovered scope established for unit testing. The unit testing is white- box oriented, and step can be conducted in parallel for multiple components. The modular interface is tested to ensure that information properly flows into and out of the program unit under test. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that all statements in a module have been executed at least once. Finally, all error handling paths are tested.

Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. Selective testing of execution paths is an essential task during the unit test. Good design dictates that error conditions be anticipated and error handling paths set up to reroute or cleanly terminate processing when an error does occur.

Boundary testing is the last task of unit testing step. Software often fails at its boundaries.

2.8.2.2 Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop. After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover differences in program structures were removed and a unique program structure was evolved.

2.8.2.3 Validation Testing or System Testing

This is the final step in testing. In this the entire system was tested as a whole with all forms, code, modules and class modules. This form of testing is popularly known as Black Box testing or System tests.

Black Box testing method focuses on the functional requirements of the software. That is, Black Box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

Black Box testing attempts to find errors in the following categories; incorrect or missing functions, interface errors, errors in data structures or external data access, performance errors and initialization errors and termination errors.

2.8.2.4 Output Testing or User Acceptance Testing

The system considered is tested for user acceptance; here it should satisfy the firm's need. The software should keep in touch with perspective system; user at the time of developing and making changes whenever required. This done with respect to the following points:

- Input Screen Designs,
- Output Screen Designs,

The above testing is done taking various kinds of test data. Preparation of test data plays a vital role in the system testing. After preparing the test data, the system under study is tested using that test data. While testing the system by which test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

2.8.3 Test Case

Test Case 1

Project Name: Eventor Event Planner-Event Management System					
Login Test Case					
Test Case ID: Fun_1		Test Designed By: Sarath Gopalakrishnan			
Test Priority(Low/Medium/High): High		Test Designed Date: 18-06-2021			
Module Name: Login Screen		Test Executed By : Ms. Gloriya Mathew			
Test Title : Verify login with valid username and password		Test Execution Date: 20-06-2021			
Description: Test the Login Page					
Pre-Condition : User has valid user name and password					
Step	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/Fail)
1	Navigation to Login Page		Login Page should be	Login page displayed	Pass
2	Provide Valid User name	User Name: manikumar34@gmail.com	User should be able to Login	User Logged in and navigated to User Dashboard with records	Pass
3	Provide Valid Password	Password: Manikumar@12			
4	Click on Login button				
5	Provide Invalid User Name or password	User Name: 13@gmail Password: mani	User should not be able to Login	Message for enter valid user or password displayed	Pass
6	Provide Null User Name or Password	User Name: null Password: null			
7	Click on Login button				
Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database					

Test Case 2					
Project Name: Eventor Event Planner-Event Management System					
User Registration Test Case					
Test Case ID:Fun_2			Test Designed By: Sarath Gopalakrishnan		
Test Priority(Low/Medium/High):Medium			Test Designed Date: 18-06-2021		
Module Name: User Registration			Test Executed By : Ms. Gloriya Mathew		
Test Title : Verify new user registration			Test Execution Date: 20-06-2021		
Description: Test the User registration Page					
Pre-Condition : User should not be already registered					
Step	Test Step	Test Data	Expected Result	Actual Result	Status(Pass/Fail)
1	Navigation to Registration Page		RegistrationPage should be displayed	Registration page displayed	Pass
2	Provide null information	Name: null	Validation message should be displayed	*Mandatory field message displayed	Pass
3	Provide Valid Details of user	All the mandatory registration details of the user	User should be able to register successfully	Registration successful and home page displayed	
4	Click on Register button				Pass
Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database					

2.9 Implementation

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later.

Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one.

At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after thorough testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover.

The implementation state involves the following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.

2.9.1 Implementation Procedures

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to ensure that the resistance does not build up, as one has to make sure that:

- The active user must be aware of the benefits of using the new system.
- Their confidence in the software is built up.
- Proper guidance is imparted to the user so that he is comfortable in using the application.

Before going ahead and viewing the system, the user must know that for viewing the result, the server program should be running in the server. If the server object is not up running on the server, the actual process won't take place.

2.9.2 User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

2.9.3 Operational Document

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered. It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy.

2.9.4 System Maintenance

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After

a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes".

2.10 Conclusion and Future Enhancements

2.10.1 Future Enhancement

- This application can be easily implemented under various situations.
- We can add new features as and when required.
- Reusability is possible as and when require in this application.
- There is flexibility in all the modules.
- It can be implemented in any server in the internet providing an easy access to the users.
- In future we can add more modules as requirement.
- The proposed system is designed in such a way that the payment of product should be done in online mode.
- Customers can able to do advanced search options
- Customers can able to cancel order return products etc.

2.10.2 Conclusion

Computer has got clear advantage over the manual system. The computerized system is more reliable, efficient and fast at the end of the project, I can say that computer play a very crucial role in the development of the firm. All the daily reports generated by the system are to be checked by the concerned official so as to ensure that all the transaction have been put in appropriate accounts and this is tallied with new vouchers.

Computer does maximum work within minimum time. Because it is used in every field so that it provides comfort and suitability to everyone. Providing maximum facilities and comfort to users is the main goal of the system. To achieve this goal, other modern facilities relating to the computer should have to be provided.

Hence, conclude by stating that, "EVENTOR EVENT PLANNER" as a computer based

application could do many more when compare to the traditional manual method. The application is designed in such a way that any further enhancement can be done with ease. The system has the capability for easy integration with other systems. New modules can be added to the existing system with less effort. In future a new function or procedure can be easily added in the system through existing classes or even a new class can be added.

2.11 Bibliography

Books/References:

- Elias M. Award – System Analysis and Design, Galgotia Publication Pvt Ltd, Year: 2002, Edition 2.
- Pankaj Jalote - An Integral Approach to Software Engineering, Narosa Publishing House Pvt Ltd, Year: 2009, Edition 3.
- Korth, Henry f- Database System Concepts, hill book, year: 2014 Edition.
- Gary B. Shelly, Harry J. Rosenblatt, “*System Analysis and Design*”, 2009.
- Roger S Pressman, “*Software Engineering*”, 1994.
- PankajJalote, “*Software engineering: a precise approach*”, 2006.
- James lee and Brent ware Addison, “Open source web development with LAMP”, 2003.
- IEEE Std 1016 Recommended Practice for Software Design Descriptions.

Websites:

- www.w3schools.com.
- www.jquery.com.
- <http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf>.
- www.agilemodeling.com/artifacts/useCaseDiagram.html.
- <https://mdbootstrap.com/snippets/jquery/mdbootstrap>.

2.12 APPENDIX

2.12.1 SAMPLE CODE

Connection.php

```
<?php
$conn = mysqli_connect("localhost","root","","db_eventor");
if(!$conn)
{
    ?>
<script type="text/javascript">
    alert("Failed to establish connection! ");
</script>
<?php
}
?>
```

userregistrationform.php

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>User Registration Form</title>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.4.0/jquery.min.js"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/jqueryvalidate/1.19.1/jquery.validate.min.js"></script>
    <link href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
rel="stylesheet">
    <link href="bootstrap/css/bootstrap.css" rel="stylesheet">
    <link href="userregistrationformstyle.css" rel="stylesheet">
    <script src="USERVALIDATIONCODE/uservalidationcode.js"></script>
</head>
<body>
<header>
    <nav class="navbar navbar-expand-lg">
```

```

<button    class="navbar-toggler"    type="button"    data-toggle="collapse"    data-
target="#navbarcontent"    aria-controls="navbar-content"    aria-expanded="false"    aria-
label="toggle-navigation">    <span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbar-content">
<ul class="navbar-nav">
<li class="nav-item">
<a class="nav-link" href="index.php">Home</a>
</li>
<li class="nav-item">
<a class="nav-link" href="loginpage.php">Login</a>
</li>
</ul>
</div>
</nav>
<div class="box">
<div class="box-1"></div>
<div class="box-2"></div>
</div>
</header>
<section>
<h4    class="text-center">Welcome    to</h4>
</section>
<h3    class="col-12 col-xl-12 col-lg-12 text-center bg-dark text-white">User Registration
Form</h3>
<form action="action-userregister.php" style="border:1px solid #ccc" method="POST"
name="Register" onsubmit= "return validates()" >
<div class="container">
<p col-6 col-xl-6 col-lg-6>Please fill in this form to create an account.</p>
<br>
<div class="form-group col-12 col-xl-12 col-lg-12">
<label for="name"><b>Name</b></label>
<input    type="text"    class="form-control"    name="name"    id="Uname"
placeholder="Enter the Name" value=""    maxlength="50" />
</div>
<div class="form-group col-12 col-xl-12 col-lg-12">
<label for="housename"><b>House Name</b></label>

```

```

<input type="text" class="form-control" name="housename" id="Uhousename"
placeholder="Enter the House Name" value="" maxlength="50" />

</div>
<div class="form-group col-12 col-xl-12 col-lg-12">
<label for="city"><b>City/Landmark</b></label>
<input type="text" class="form-control" name="city" id = "UCityname"
placeholder="Enter the City/Landmark" value="" maxlength="50" />

</div>
<div class="form-group col-12 col-xl-12 col-lg-12">
<label for="postoffice"><b>Post Office</b></label>
<input type="text" class="form-control" name="postoffice" id="UPostoffice"
placeholder="Enter Postoffice" value="" maxlength="50" />

</div>
<div class="form-group col-12 col-xl-12 col-lg-12">
<label for="pincode"><b>Pincode</b></label>
<input type="text" class="form-control" name="pincode" id="UPincode"
placeholder="Enter the Pincode" value="" maxlength="6" />

</div>
<div class="form-group col-12 col-xl-12 col-lg-12">
<label for="district"><b>District</b></label>
<select class="form-control" id="UDistrict" name="district" >
<option value="" disabled selected>--Please Select your District--</option>
<?php include
'connection.php';

$sql="SELECT * FROM `tbl_district`";
$result= mysqli_query($conn, $sql);
while($row = mysqli_fetch_assoc($result))

{
    echo "<option value='".$row['did']."'>".$row['dname']."</option>";
}
?>
</select>
</div>
<div class="form-group col-12 col-xl-12 col-lg-12">
<label for="gender"><b>Gender</b></label>
<input type="radio" id="malec" name="gender" value="male"/> Male
<input type="radio" id="femalec" name="gender" value="female" /> Female

```

```

<input type="radio" id="otherc" name="gender" value="other" /> Other
<label id="Ugender"></label>
</div>
<div class="form-group col-12 col-xl-12 col-lg-12">
<label for="phoneno"><b>Phone Number</b></label>

<input type="text" class="form-control" name="phoneno" id="UPhone" value="" maxlength="10" placeholder="Enter the Phone Number" />
</div>
<div class="form-group col-12 col-xl-12 col-lg-12">
<label for="email"><b>Email</b></label>
<input type="text" class="form-control" id="Uemail" placeholder="Enter the EmailId" name="email" value="" />
</div>
<div class="form-group col-12 col-xl-12 col-lg-12">
<label for="password"><b>Password</b></label>
<input type="password" class="form-control" id="UPword" placeholder="Example@123" name="password" value="" maxlength="8" />
</div>
<div class="form-group col-12 col-xl-12 col-lg-12">
<label for="cpassword"><b>Confirm Password</b></label>
<input type="password" class="form-control" id="UCpword" placeholder="Re-enter the Password" name="cpassword" value="" maxlength="8"/>
</div>
<label>
<input type="checkbox" name="remember" style="margin-bottom:15px" checked="checked" >
Remember me
</label>
<p>By creating an account you agree to our <a href="#" style="color:dodgerblue">Terms & Privacy</a>.</p>
<div class="clearfix">
<button type="submit" name="submit" class="registerbtn" value="Register">Register</button>
<button type="reset" class="cancelbtn">Cancel</button>
</div>
</div>
</form>

```

```
</body>
</html>
```

action-userregister.php

```
<?php      include
'connection.php';

$name = $_POST['name'];
$housename = $_POST['housename'];
$city = $_POST['city'];
$postoffice = $_POST['postoffice'];
$pincode = $_POST['pincode'];
$dname = $_POST['district'];
$gender = $_POST['gender'];
$phoneno = $_POST['phoneno'];
$email = $_POST['email'];
$password = $_POST['password'];
$cpassword = $_POST['cpassword'];
$sql= "INSERT INTO `tbl_login`(`lid`, `email`, `password`, `role`, `status`) VALUES
('$email','$password', 'User', 4)";

mysqli_query($conn, $sql);

$lid=mysqli_insert_id($conn);

$sql = "INSERT INTO `tbl_userreg`(`userid`, `name`, `housename`, `city`, `postoffice`,
`pincode`, `did`, `gender`, `phoneno`, `lid`) VALUES
('$name','$housename','$city','$postoffice','$pincode','$dname','$gender','$phoneno','$l
id')";
if(mysqli_query($conn, $sql)){
    echo '<script type="text/javascript">';
    echo 'alert("Registered Successfully")';
    echo '</script>';
}

echo("<script>location.href='http://localhost/EVENTOR/index.php'</script>");
} else{
    echo '<script
type="text/javascript">';
    echo 'alert("Something Wrong")';
    echo '</script>';
}

mysqli_close($conn);
?>
```

Adddecoration.php

```
<?php
    include 'connection.php';
    session_start();
    if(ISSET($_SESSION["lid"])){
        $s_id=$_SESSION["lid"];
    }
    else{
        header("location:index.php");
    }
?>

<!DOCTYPE html>
<html>

<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Eventor Event Planner</title>
    <!-- Tell the browser to be responsive to screen width -->
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- Font Awesome -->
    <link rel="stylesheet" href="../../plugins/fontawesome-free/css/all.min.css">
    <!-- Ionicons -->
    <link rel="stylesheet"
    href="https://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css">
    <!-- DataTables -->
    <link rel="stylesheet" href="../../plugins/datatables-bs4/css/dataTables.bootstrap4.min.css">
    <link rel="stylesheet" href="../../plugins/datatables-responsive/css/responsive.bootstrap4.min.css">
    <!-- Theme style -->
```

```
<link rel="stylesheet" href="../../dist/css/adminlte.min.css">
<!-- Google Font: Source Sans Pro -->
<link href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700"
rel="stylesheet">
<script src="ADDWORKTYPE/addworktype.js"></script>

</head>
<body class="hold-transition sidebar-mini">
<div class="wrapper">
<!-- Navbar -->
<nav class="main-header navbar navbar-expand navbar-white navbar-light">
<!-- Left navbar links -->
<ul class="navbar-nav">
<li class="nav-item">
<a class="nav-link" data-widget="pushmenu" href="#" role="button"><i class="fas fa-bars"></i></a>
</li>
<li class="nav-item d-none d-sm-inline-block">
<a href="../../index3.html" class="nav-link">Home</a>
</li>
<li class="nav-item d-none d-sm-inline-block">
<a href="#" class="nav-link">Contact</a>
</li>
</ul>

<!-- SEARCH FORM -->
<form class="form-inline ml-3">
<div class="input-group input-group-sm">
<input class="form-control form-control-navbar" type="search" placeholder="Search" aria-
label="Search">
<div class="input-group-append">
<button class="btn btn-navbar" type="submit">
<i class="fas fa-search"></i>
```

```
</button>
</div>
</div>
</form>

<!-- Right navbar links -->
<ul class="navbar-nav ml-auto">
  <!-- Messages Dropdown Menu -->
  <li class="nav-item dropdown">
    <a class="nav-link" data-toggle="dropdown" href="#">
      <i class="far fa-comments"></i>
      <span class="badge badge-danger navbar-badge">3</span>
    </a>
    <div class="dropdown-menu dropdown-menu-lg dropdown-menu-right">
      <a href="#" class="dropdown-item">
        <!-- Message Start -->
        <div class="media">
          
          <div class="media-body">
            <h3 class="dropdown-item-title">
              Brad Diesel
              <span class="float-right text-sm text-danger"><i class="fas fa-star"></i></span>
            </h3>
            <p class="text-sm">Call me whenever you can...</p>
            <p class="text-sm text-muted"><i class="far fa-clock mr-1"></i> 4 Hours Ago</p>
          </div>
        </div>
        <!-- Message End -->
      </a>
      <div class="dropdown-divider"></div>
      <a href="#" class="dropdown-item">
```

```
<!-- Message Start -->
<div class="media">
    
    <div class="media-body">
        <h3 class="dropdown-item-title">
            John Pierce
            <span class="float-right text-sm text-muted"><i class="fas fa-star"></i></span>
        </h3>
        <p class="text-sm">I got your message bro</p>
        <p class="text-sm text-muted"><i class="far fa-clock mr-1"></i> 4 Hours Ago</p>
    </div>
</div>
<!-- Message End -->
</a>
<div class="dropdown-divider"></div>
<a href="#" class="dropdown-item">
    <!-- Message Start -->
    <div class="media">
        
        <div class="media-body">
            <h3 class="dropdown-item-title">
                Nora Sylvester
                <span class="float-right text-sm text-warning"><i class="fas fa-star"></i></span>
            </h3>
            <p class="text-sm">The subject goes here</p>
            <p class="text-sm text-muted"><i class="far fa-clock mr-1"></i> 4 Hours Ago</p>
        </div>
    </div>
    <!-- Message End -->
</a>
```

```
<div class="dropdown-divider"></div>
<a href="#" class="dropdown-item dropdown-footer">See All Messages</a>
</div>
</li>
<!-- Notifications Dropdown Menu -->
<li class="nav-item dropdown">
  <a class="nav-link" data-toggle="dropdown" href="#">
    <i class="far fa-bell"></i>
    <span class="badge badge-warning navbar-badge">15</span>
  </a>
  <div class="dropdown-menu dropdown-menu-lg dropdown-menu-right">
    <span class="dropdown-item dropdown-header">15 Notifications</span>
    <div class="dropdown-divider"></div>
    <a href="#" class="dropdown-item">
      <i class="fas fa-envelope mr-2"></i> 4 new messages
      <span class="float-right text-muted text-sm">3 mins</span>
    </a>
    <div class="dropdown-divider"></div>
    <a href="#" class="dropdown-item">
      <i class="fas fa-users mr-2"></i> 8 friend requests
      <span class="float-right text-muted text-sm">12 hours</span>
    </a>
    <div class="dropdown-divider"></div>
    <a href="#" class="dropdown-item">
      <i class="fas fa-file mr-2"></i> 3 new reports
      <span class="float-right text-muted text-sm">2 days</span>
    </a>
    <div class="dropdown-divider"></div>
    <a href="#" class="dropdown-item dropdown-footer">See All Notifications</a>
  </div>
</li>
<li class="nav-item">
```

```
<a class="nav-link" data-widget="control-sidebar" data-slide="true" href="#" role="button">  
    <i class="fas fa-th-large"></i>  
</a>  
</li>  
</ul>  
</nav>  
<!-- /.navbar -->
```

```
<?php
```

```
include ('includes/sidebar.php');  
?>  
<!-- Content Wrapper. Contains page content -->  
<div class="content-wrapper">  
    <!-- Content Header (Page header) -->  
    <section class="content-header">  
        <div class="container-fluid">  
            <div class="row mb-2">  
                <div class="col-sm-6">  
                    <h1>DataTables</h1>  
                </div>  
                <div class="col-sm-6">  
                    <ol class="breadcrumb float-sm-right">  
                        <li class="breadcrumb-item"><a href="#">Home</a></li>  
                        <li class="breadcrumb-item active">DataTables</li>  
                    </ol>  
                </div>  
            </div><!-- /.container-fluid -->  
        </section>
```

```
<div class="modal fade" id="adddecoration" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">

        <h5 class="modal-title" id="exampleModalLabel">Add Decoration Details</h5>

        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <form action="action-adddecorationdetails.php" style="border:1px solid #ccc"
method="POST" enctype="multipart/form-data" name="Register" id="Register"
onsubmit="return validates()">

        <div class="modal-body">

          <div class="form-group">
            <label for="eventname"><b>Event</b></label>
            <select class="form-control" id="UEvent" name="eventname" required="required">
              <option value="" disabled selected>--Please Select Event-- </option>
            <?php
              $sql="SELECT * FROM `tbl_addeventtype`";
              $result= mysqli_query($conn, $sql);
              while($row = mysqli_fetch_assoc($result))
              {
                echo "<option value='".$row['eventtypeid']."'>".$row['eventname']."</option>";
              }
            ?>
```

```
</select>
</div>

    <div class="form-group">
        <label for="image">Image</label>
        <input type="file" class="form-control-file" name="image" id="image">
    </div>

    <div class="form-group">
        <label for="description"><b>Decoration Details</b></label><br>
        <textarea class="form-control" name="description" id="Udescription" rows="5" cols="50" maxlength="100" placeholder="Please enter the details" required="required"></textarea>
    </div>

    <div class="form-group">
        <label for="amount"><b>Amount</b></label>
        <input type="number" class="form-control" name="amount" id="UAmount" value="" maxlength="8" placeholder="Enter the Amount" required="required" />
    </div>
</div>

<div class="modal-footer">
    <button type="submit" name="submit" class="btn btn-outline-primary" value="Register">Save</button>
    <button type="reset" name="cancel" class="btn btn-outline-danger">Cancel</button>
</div>
</form>
</div>
</div>

<div class="container-fluid">
<div class="card shawdow mb-4">
    <div class="card-header py-3">
        <center>
            <h3 class="m-0 font-weight-bold text-primary"> ADD DECORATION DETAILS </h3>
            <h6>
```

```

<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#adddecoration">
    Add Decoration Details
</button>
</h6>
</center>
</div>
</div>
</div>

<!-- Main content --&gt;
&lt;!-- /.card --&gt;
&lt;div class="container-fluid"&gt;
&lt;div class="row"&gt;
&lt;div class="col-12"&gt;
&lt;!-- Start DataTales Example --&gt;
&lt;div class="card shadow mb-4"&gt;
&lt;div class="card-header py-3"&gt;
&lt;!-- /.card --&gt;
&lt;h6 class="m-0 font-weight-bold text-primary"&gt;Added Decoration Details&lt;/h6&gt;
&lt;/div&gt;
&lt;div class="card-body"&gt;
    &lt;div class="table-responsive"&gt;
        &lt;?php
            $query = "SELECT `tbl_adddecorationdetails`.* , `tbl_addeventtype` . `eventname` FROM
            `tbl_adddecorationdetails` INNER JOIN `tbl_companyreg` ON
            `tbl_companyreg.lid` = `tbl_adddecorationdetails.lid` INNER JOIN `tbl_addeventtype` ON
            `tbl_addeventtype.eventtypeid` = `tbl_adddecorationdetails.eventtypeid` INNER JOIN `tbl_login` ON
            `tbl_login.lid` = `tbl_adddecorationdetails.lid` WHERE `tbl_adddecorationdetails.status` = 0 AND
            `tbl_adddecorationdetails.lid` = $s_id AND `tbl_companyreg.lid` = $s_id";
            $query_run = mysqli_query($conn, $query);
        ?&gt;
        &lt;table class="table table-bordered" id="dataTable" width="100%" cellspacing="0"&gt;
            &lt;thead&gt;
</pre>

```

```
<tr>
    <th> Event Name</th>
    <th> Decoration Image</th>
    <th> Decoration Details</th>
    <th> Amount</th>
    <th> EDIT</th>
    <th> DELETE</th>
</tr>
</thead>
<tbody>
<?php
if(mysqli_num_rows($query_run) > 0)
{
    while($row = mysqli_fetch_assoc($query_run))
    {
        ?>
<tr>
    <td><?php echo $row['eventname']; ?></td>
    <td> <?php echo ''></td>
    <td><?php echo $row['description']; ?></td>
    <td><?php echo $row['amount']; ?></td>
        <td>
            <form action="decorationdetailseditform.php" method="post">
                <input type="hidden" name="edit_decorationdetailid" value="<?php echo $row['decorationid']; ?>">
                <button type="submit" name="edit_btn" class="btn btn-success">EDIT</button>
            </form>
        </td>
    <td>
        <form action="deletedecorationdetails.php" method="post">
```

```
<input type="hidden" name="delete_decorationdetailid" value="php echo<br/$row['decorationid']; ?>">  
    <button type="submit" name="delete_btn" class="btn btn-danger">  
DELETE</button>  
    </form>  
  </td>  
  
</tr>  
<?php  
}  
}  
else {  
    echo "No Record Found";  
}  
?  
</tbody>  
</table>  
</div>  
</div>  
</div>  
<!-- /.card -->  
<!-- End DataTales Example -->  
<!-- Main content -->  
    <!-- /.card -->  
    <!-- Start DataTales Example -->  
<div class="card shadow mb-4">  
    <div class="card-header py-3">  
        <h6 class="m-0 font-weight-bold text-primary">Deleted Decoration Details</h6>  
    </div>  
    <div class="card-body">  
        <div class="table-responsive">  
            <?php
```

```

$query ="SELECT `tbl_adddecorationdetails`.* ,`tbl_addeventtype` .`eventname` FROM
tbl_adddecorationdetails INNER JOIN tbl_companyreg ON
tbl_companyreg.lid=tbl_adddecorationdetails.lid INNER JOIN tbl_addeventtype ON
tbl_addeventtype.eventtypeid=tbl_adddecorationdetails.eventtypeid INNER JOIN tbl_login ON
tbl_login.lid=tbl_adddecorationdetails.lid WHERE tbl_adddecorationdetails.`status`=2 and
tbl_adddecorationdetails.lid=$s_id and tbl_companyreg.lid=$s_id";

$query_run = mysqli_query($conn, $query);

?>

<table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">
<thead>
<tr>
    <th> Event Name</th>
    <th> Decoration Image</th>
    <th> Decoration Details</th>
    <th> Amount</th>
    <th> Action</th>
</tr>
</thead>
<tbody>
<?php
if(mysqli_num_rows($query_run) > 0)
{
while($row = mysqli_fetch_assoc($query_run))
{
?>
<tr>
    <td><?php echo $row['eventname']; ?></td>
    <td> <?php echo ''></td>
    <td><?php echo $row['description']; ?></td>
    <td><?php echo $row['amount']; ?></td>

```

```
<td>
    <form action="deletedecorationdetailsadd.php" method="post">
        <input type="hidden" name="delete_decorationdetailid" value="<?php echo
$row['decorationid']; ?>">
        <button type="submit" name="delete_btn" class="btn btn-
primary">ADD</button>
    </form>
</td>
</tr>
<?php
}
}
else {
    echo "No Record Found";
}
?>
</tbody>
</table>
</div>
</div>
</div>
<!-- /.card -->

<!-- Start DataTales Example -->
<div class="card shadow mb-4">
    <div class="card-header py-3">
        <h6 class="m-0 font-weight-bold text-primary">Approved Decoration
Details</h6>
    </div>
    <div class="card-body">
        <div class="table-responsive">
            <?php
```

```

$query ="SELECT `tbl_adddecorationdetails`.* ,`tbl_addeventtype` .`eventname` FROM
tbl_adddecorationdetails INNER JOIN tbl_companyreg ON
tbl_companyreg.lid=tbl_adddecorationdetails.lid INNER JOIN tbl_addeventtype ON
tbl_addeventtype.eventtypeid=tbl_adddecorationdetails.eventtypeid INNER JOIN tbl_login ON
tbl_login.lid=tbl_adddecorationdetails.lid WHERE tbl_adddecorationdetails.`status`=1 and
tbl_adddecorationdetails.lid=$s_id and tbl_companyreg.lid=$s_id";

$query_run = mysqli_query($conn, $query);

?>

<table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">
<thead>
<tr>
<th> SI No </th>
<th> Event Name</th>
<th> Decoration Image</th>
<th> Decoration Details</th>
<th> Amount</th>
</tr>
</thead>

<tbody>
<?php
if(mysqli_num_rows($query_run) > 0)
{
while($row = mysqli_fetch_assoc($query_run))
{
?>
<tr>
<td><?php echo $row['decorationid']; ?></td>
<td><?php echo $row['eventname']; ?></td>
<td> <?php echo ''></td>
<td><?php echo $row['description']; ?></td>
<td><?php echo $row['amount']; ?></td>
</tr>
}
}
}

```

```
<?php
}
}
else {
    echo "No Record Found";
}
?>
</tbody>
</table>
</div>
</div>
</div>
</div>
<!-- /.col -->
</div>
<!-- /.row -->
</div>
<!-- /.container-fluid -->
</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->
<footer class="main-footer">
    <div class="float-right d-none d-sm-block">
        <b>Version</b> 3.0.5
    </div>
    <strong>Copyright &copy; 2014-2019 <a href="http://adminlte.io">Eventor Event
Planner</a>.</strong> All rights
reserved.
</footer>

<!-- Control Sidebar -->
```

```
<aside class="control-sidebar control-sidebar-dark">
    <!-- Control sidebar content goes here -->
</aside>
<!-- /.control-sidebar -->
</div>
<!-- ./wrapper -->
<!-- jQuery -->
<script src="../../plugins/jquery/jquery.min.js"></script>
<!-- Bootstrap 4 -->
<script src="../../plugins/bootstrap/js/bootstrap.bundle.min.js"></script>
<!-- DataTables -->
<script src="../../plugins/datatables/jquery.dataTables.min.js"></script>
<script src="../../plugins/datatables-bs4/js/dataTables.bootstrap4.min.js"></script>
<script src="../../plugins/datatables-responsive/js/dataTables.responsive.min.js"></script>
<script src="../../plugins/datatables-responsive/js/responsive.bootstrap4.min.js"></script>
<!-- AdminLTE App -->
<script src="../../dist/js/adminlte.min.js"></script>
<!-- AdminLTE for demo purposes -->
<script src="../../dist/js/demo.js"></script>
<!-- page script -->
<!-- Page level plugins -->
<script src="vendor/datatables/jquery.dataTables.min.js"></script>
<script src="vendor/datatables/dataTables.bootstrap4.min.js"></script>
<!--Page level custom scripts -->
<script src="js/demo/datatables-demo.js"></script>
<script>
    $(document).ready(function(){
        $('#search').keyup(function(){
            search_table($(this).val());
        });
        function search_table(value){
            $('#employee_table tr').each(function(){

```

```
var found = 'false';
$(this).each(function(){
    if($(this).text().toLowerCase().indexOf(value.toLowerCase()) >= 0)
    {
        found = 'true';
    }
});
if(found == 'true')
{
    $(this).show();
}
else
{
    $(this).hide();
})
});
});

</script>
<script>
$(function () {
    $('#example1').DataTable({
        "responsive": true,
        "autoWidth": false,
    });
    $('#example2').DataTable({
        "paging": true,
        "lengthChange": false,
        "searching": false,
        "ordering": true,
        "info": true,
        "autoWidth": false,
    });
});
```

```

    "responsive": true,
  });
}

```

```

</script>
</body>
</html>

```

action-decorationdetails.php

```

<?php
include 'connection.php';
session_start();
if(ISSET($_SESSION["lid"])){
  $s_id=$_SESSION["lid"];
$eventname = $_POST['eventname'];
$image = $_FILES["image"]['name'];
getDescription = $_POST['description'];
$amount= $_POST['amount'];

$sql= "INSERT INTO `tbl_adddecorationdetails`(`decorationid`, `lid`, `eventtypeid`, `image`, `des
cription`, `amount`, `status`) VALUES (',$s_id','$eventname','$image','$description','$amount','
')";
// mysqli_query($conn, $sql);
if(mysqli_query($conn, $sql))
{
  move_uploaded_file($_FILES["image"]["tmp_name"],"upload/".$_FILES["image"]["name"]);
  echo '<script type="text/javascript">';
  echo 'alert("Added Successfully")';
  echo '</script>';
  echo ("<script>location.href='http://localhost/EVENTORSYSTEM/company/pages/forms/addd
ecorationdetails.php'</script>");
}
else
{
  echo '<script type="text/javascript">';
  echo 'alert("Not Added ")';
  echo '</script>';
}

mysqli_close($conn);
?>

```

Approvevenue.php

```
<?php
    include 'connection.php';

?>

<!DOCTYPE html>
<html>

<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>Eventor Event Planner</title>
    <!-- Tell the browser to be responsive to screen width -->
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- Font Awesome -->
    <link rel="stylesheet" href="../../plugins/fontawesome-free/css/all.min.css">
    <!-- Ionicons -->
    <link rel="stylesheet"
        href="https://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css">
    <!-- DataTables -->
    <link rel="stylesheet" href="../../plugins/datatables-bs4/css/dataTables.bootstrap4.min.css">
    <link rel="stylesheet" href="../../plugins/datatables-responsive/css/responsive.bootstrap4.min.css">
    <!-- Theme style -->
    <link rel="stylesheet" href="../../dist/css/adminlte.min.css">
    <!-- Google Font: Source Sans Pro -->
    <link
        href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700"
        rel="stylesheet">
    <script src="ADDWORKTYPE/addworktype.js"></script>
```

```
</head>
<body class="hold-transition sidebar-mini">
<div class="wrapper">
<!-- Navbar -->
<nav class="main-header navbar navbar-expand navbar-white navbar-light">
<!-- Left navbar links -->
<ul class="navbar-nav">
<li class="nav-item">
<a class="nav-link" data-widget="pushmenu" href="#" role="button"><i class="fas fa-bars"></i></a>
</li>
<li class="nav-item d-none d-sm-inline-block">
<a href="../../index3.html" class="nav-link">Home</a>
</li>
<li class="nav-item d-none d-sm-inline-block">
<a href="#" class="nav-link">Contact</a>
</li>
</ul>

<!-- SEARCH FORM -->
<form class="form-inline ml-3">
<div class="input-group input-group-sm">
<input class="form-control form-control-navbar" type="search"
placeholder="Search" aria-label="Search">
<div class="input-group-append">
<button class="btn btn-navbar" type="submit">
<i class="fas fa-search"></i>
</button>
</div>
</div>
</form>
```

```
<!-- Right navbar links -->
<ul class="navbar-nav ml-auto">
    <!-- Messages Dropdown Menu -->
    <li class="nav-item dropdown">
        <a class="nav-link" data-toggle="dropdown" href="#">
            <i class="far fa-comments"></i>
            <span class="badge badge-danger navbar-badge">3</span>
        </a>
        <div class="dropdown-menu dropdown-menu-lg dropdown-menu-right">
            <a href="#" class="dropdown-item">
                <!-- Message Start -->
                <div class="media">
                    
                    <div class="media-body">
                        <h3 class="dropdown-item-title">
                            Brad Diesel
                            <span class="float-right text-sm text-danger"><i class="fas fa-star"></i></span>
                        </h3>
                        <p class="text-sm">Call me whenever you can...</p>
                        <p class="text-sm text-muted"><i class="far fa-clock mr-1"></i> 4 Hours Ago</p>
                    </div>
                </div>
                <!-- Message End -->
            </a>
            <div class="dropdown-divider"></div>
            <a href="#" class="dropdown-item">
                <!-- Message Start -->
                <div class="media">
                    
                    <div class="media-body">
```

```
<h3 class="dropdown-item-title">
    John Pierce
    <span class="float-right text-sm text-muted"><i class="fas fa-star"></i></span>
</h3>
<p class="text-sm">I got your message bro</p>
<p class="text-sm text-muted"><i class="far fa-clock mr-1"></i> 4 Hours Ago</p>
</div>
</div>
<!-- Message End -->
</a>
<div class="dropdown-divider"></div>
<a href="#" class="dropdown-item">
    <!-- Message Start -->
    <div class="media">
        
        <div class="media-body">
            <h3 class="dropdown-item-title">
                Nora Silvester
                <span class="float-right text-sm text-warning"><i class="fas fa-star"></i></span>
            </h3>
            <p class="text-sm">The subject goes here</p>
            <p class="text-sm text-muted"><i class="far fa-clock mr-1"></i> 4 Hours Ago</p>
        </div>
    </div>
    <!-- Message End -->
</a>
<div class="dropdown-divider"></div>
<a href="#" class="dropdown-item dropdown-footer">See All Messages</a>
</div>
```

```
</li>
<!-- Notifications Dropdown Menu -->
<li class="nav-item dropdown">
  <a class="nav-link" data-toggle="dropdown" href="#">
    <i class="far fa-bell"></i>
    <span class="badge badge-warning navbar-badge">15</span>
  </a>
  <div class="dropdown-menu dropdown-menu-lg dropdown-menu-right">
    <span class="dropdown-item dropdown-header">15 Notifications</span>
    <div class="dropdown-divider"></div>
    <a href="#" class="dropdown-item">
      <i class="fas fa-envelope mr-2"></i> 4 new messages
      <span class="float-right text-muted text-sm">3 mins</span>
    </a>
    <div class="dropdown-divider"></div>
    <a href="#" class="dropdown-item">
      <i class="fas fa-users mr-2"></i> 8 friend requests
      <span class="float-right text-muted text-sm">12 hours</span>
    </a>
    <div class="dropdown-divider"></div>
    <a href="#" class="dropdown-item">
      <i class="fas fa-file mr-2"></i> 3 new reports
      <span class="float-right text-muted text-sm">2 days</span>
    </a>
    <div class="dropdown-divider"></div>
    <a href="#" class="dropdown-item dropdown-footer">See All Notifications</a>
  </div>
</li>
<li class="nav-item">
  <a class="nav-link" data-widget="control-sidebar" data-slide="true" href="#" role="button">
    <i class="fas fa-th-large"></i>
  </a>
```

```
</li>
</ul>
</nav>
<!-- /.navbar -->

<?php

include ('includes/sidebar.php');
?>
<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
<!-- Content Header (Page header) -->
<section class="content-header">
<div class="container-fluid">
<div class="row mb-2">
<div class="col-sm-6">
<h1>DataTables</h1>
</div>
<div class="col-sm-6">
<ol class="breadcrumb float-sm-right">
<li class="breadcrumb-item"><a href="#">Home</a></li>
<li class="breadcrumb-item active">DataTables</li>
</ol>
</div>
</div>
</div><!-- /.container-fluid -->
</section>

<!-- Main content -->
```

```

<!-- /.card -->
<div class="container-fluid">
<div class="row">
<div class="col-12">
<!-- Start DataTales Example -->
<div class="card shadow mb-4">
    <div class="card-header py-3">
        <h6 class="m-0 font-weight-bold text-primary">Pending Venue Details</h6>
    </div>

    <div class="card-body">
        <div class="table-responsive">
            <?php

                $query = "SELECT `tbl_addvenuedetails`.* ,tbl_companyreg.companyname
                FROM tbl_addvenuedetails INNER JOIN `tbl_companyreg` ON
                `tbl_companyreg`.`lid` = `tbl_addvenuedetails`.`lid` WHERE tbl_addvenuedetails.status=0
                ";
                $query_run = mysqli_query($conn, $query);

            ?>
            <table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">
                <thead>
                    <tr>
                        <th> Sl No </th>
                        <th> Company Name</th>
                        <th> Venue Place</th>
                        <th> Facility</th>
                        <th>Image</th>
                        <th> Seating Capacity</th>
                
```

```
<th> Standing Capacity</th>
<th>Venue Details</th>

<th> Amount</th>

<th>Status</th>
<th>Action</th>
<!-- <th>Status</th>
<th>Approve</th>
<th>Reject</th> -->
</tr>
</thead>

<tbody>
<?php

if(mysqli_num_rows($query_run) > 0)
{
while($row = mysqli_fetch_assoc($query_run))
{
?>
<tr>
<td><?php echo $row['venueid']; ?></td>
<td><?php echo $row['companyname']; ?></td>

<td><?php echo $row['venue']; ?></td>

<td><?php echo $row['facility']; ?></td>
<td> <?php echo ''?>
</td>
```

```

<td><?php echo $row['seatingcapacity']; ?></td>

<td><?php echo $row['standingcapacity']; ?></td>
<td><?php echo $row['description']; ?></td>
<td><?php echo $row['amount']; ?></td>
<td>

<?php
    if($row['status']==1)
    {
        echo "<p style='color:green' id=str1".$row['venueid'].">>Approved</p>";
    }
    else{
        echo "<p style='color:red' id=str2".$row['venueid'].">>Pending</p>";
    }
?>

</td>
<td>

<select onchange="active_Disactive_District(this.value,<?php echo $row['venueid'];?>)">
    <option value="" disabled selected>--Action--</option>
    <option value="1">Active</option>
    <option value="0">Inactive</option>
</select>
</td>

</tr>
<?php

```

```
        }
    }

else {
    echo "No Record Found";
}
?>
</tbody>

</table>
</div>
</div>
</div>
<div class="card shadow mb-4">
    <div class="card-header py-3">
        <h6 class="m-0 font-weight-bold text-primary">Approved Venue
Details</h6>
    </div>

    <div class="card-body">
        <div class="table-responsive">
            <?php

                $query = "SELECT `tbl_addvenuedetails`.* ,tbl_companyreg.companyname
FROM tbl_addvenuedetails INNER JOIN `tbl_companyreg` ON
`tbl_companyreg`.`lid` = `tbl_addvenuedetails`.`lid` WHERE
tbl_addvenuedetails.status=1";
                $query_run = mysqli_query($conn, $query);

            ?>
```

```
<table class="table table-bordered" id="dataTable" width="100%"  
cellspacing="0">  
    <thead>  
        <tr>  
            <th> Sl No </th>  
            <th> Company Name</th>  
            <th> Venue Place</th>  
            <th> Facility</th>  
            <th>Image</th>  
            <th> Seating Capacity</th>  
            <th> Standing Capacity</th>  
            <th>Venue Details</th>  
  
            <th> Amount</th>  
  
            <th>Status</th>  
            <th>Action</th>  
            <!-- <th>Status</th>  
            <th>Approve</th>  
            <th>Reject</th> -->  
        </tr>  
    </thead>  
  
    <tbody>  
        <?php  
  
        if(mysqli_num_rows($query_run) > 0)  
        {  
            while($row = mysqli_fetch_assoc($query_run))  
            {  
                ?>
```

```

<tr>
<td><?php echo $row['venueid']; ?></td>
<td><?php echo $row['companyname']; ?></td>

<td><?php echo $row['venue']; ?></td>

<td><?php echo $row['facility']; ?></td>
<td> <?php echo ''?
</td>
<td><?php echo $row['seatingcapacity']; ?></td>

<td><?php echo $row['standingcapacity']; ?></td>
<td><?php echo $row['description']; ?></td>
<td><?php echo $row['amount']; ?></td>
<td>

<?php
if($row['status']==1)
{
    echo "<p style='color:green' id=str1".$row['venueid'].">>Approved</p>";
}
else{
    echo "<p style='color:red' id=str2".$row['venueid'].">>Pending</p>";
}
?>
</td>
<td>

<select onchange="active_Disactive_District(this.value,<?php echo $row['venueid'];?>)">

```

```
<option value="" disabled selected>--Action--</option>
<option value="1">Active</option>
<option value="0">Inactive</option>
</select>
</td>

</tr>
<?php
}
}

else {
    echo "No Record Found";
}
?>

</tbody>
</table>
</div>
</div>
</div>
<!-- /.card -->
<!-- End DataTales Example -->

<!-- Main content -->
<!-- /.card -->

</div>
<!-- /.col -->
</div>
<!-- /.row -->
</div>
<!-- /.container-fluid -->
```

```
</section>
<!-- /.content -->
</div>
<!-- /.content-wrapper -->
<footer class="main-footer">
  <div class="float-right d-none d-sm-block">
    <b>Version</b> 3.0.5
  </div>
  <strong>Copyright &copy; 2014-2019 <a href="http://adminlte.io">Eventor Event
Planner</a>.</strong> All rights
reserved.
</footer>
<!-- Control Sidebar -->
<aside class="control-sidebar control-sidebar-dark">
  <!-- Control sidebar content goes here -->
</aside>
<!-- /.control-sidebar -->
</div>
<!-- ./wrapper -->
<!-- jQuery -->
<script src="../../plugins/jquery/jquery.min.js"></script>
<!-- Bootstrap 4 -->
<script src="../../plugins/bootstrap/js/bootstrap.bundle.min.js"></script>
<!-- DataTables -->
<script src="../../plugins/datatables/jquery.dataTables.min.js"></script>
<script src="../../plugins/datatables-bs4/js/dataTables.bootstrap4.min.js"></script>
<script src="../../plugins/datatables-responsive/js/dataTables.responsive.min.js"></script>
<script src="../../plugins/datatables-responsive/js/dataTables.responsive.bootstrap4.min.js"></script>
<!-- AdminLTE App -->
<script src="../../dist/js/adminlte.min.js"></script>
<!-- AdminLTE for demo purposes -->
<script src="../../dist/js/demo.js"></script>
<!-- page script -->
```

```
<!-- Page level plugins -->
<script src="vendor/datatables/jquery.dataTables.min.js"></script>
<script src="vendor/datatables/dataTables.bootstrap4.min.js"></script>
<!--Page level custom scripts -->
<script src="js/demo/datatables-demo.js"></script>
<script type="text/javascript">
    function active_Disactive_District(val,venueid)
    {
        $.ajax({
            type:'post',
            url:'action-venueapprove.php',
            data:{ val:val,venueid:venueid },
            success: function(result){
                if(result==1){
                    $('#str1'+venueid).html('Approved');
                }else{
                    $('#str2'+venueid).html('Pending');
                }
            }
        });
    }

    $(document).ready(function(){
        $('#search').keyup(function(){
            search_table($(this).val());
        });
        function search_table(value){
            $('#employee_table tr').each(function(){
                var found = 'false';
                $(this).each(function(){
                    if($(this).text().toLowerCase().indexOf(value.toLowerCase()) >= 0)
                    {
                        found = 'true';
                    }
                })
            })
        }
    })
}
```

```
});  
if(found == 'true')  
{  
    $(this).show();  
}  
else  
{  
    $(this).hide();  
}  
});  
});  
</script>
```

```
<script>  
$(function () {  
    $("#example1").DataTable({  
        "responsive": true,  
        "autoWidth": false,  
    });  
    $("#example2").DataTable({  
        "paging": true,  
        "lengthChange": false,  
        "searching": false,  
        "ordering": true,  
        "info": true,  
        "autoWidth": false,  
        "responsive": true,  
    });  
});  
</script>  
</body>  
</html>
```

2.10.2 SCREENSHOTS

Home Page

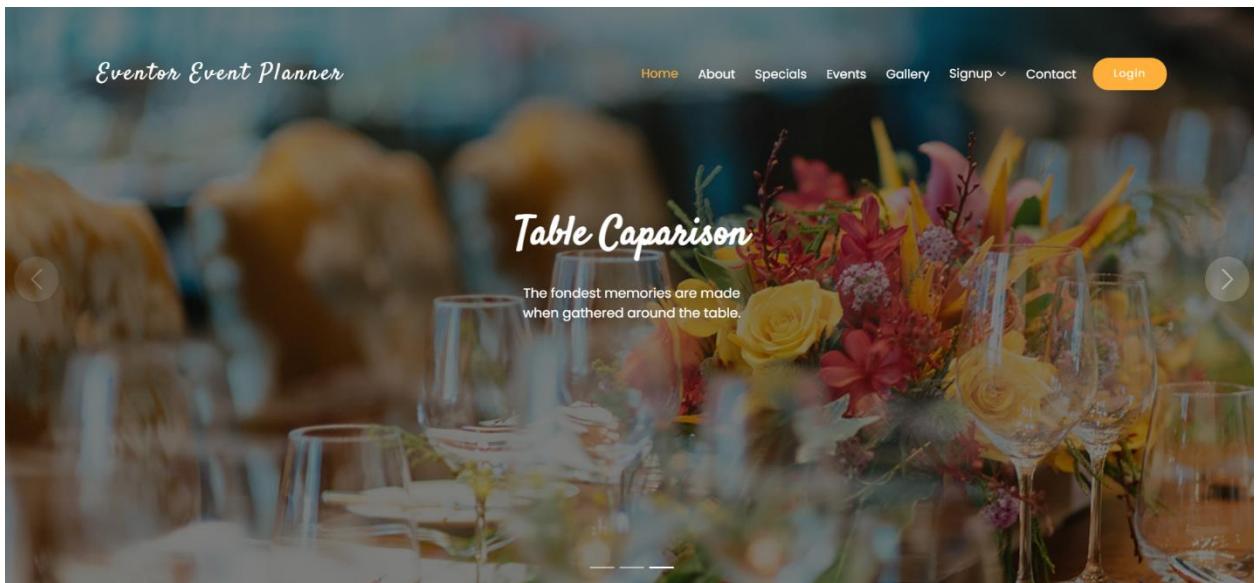


Fig 1: Home Page

LOGIN PAGE

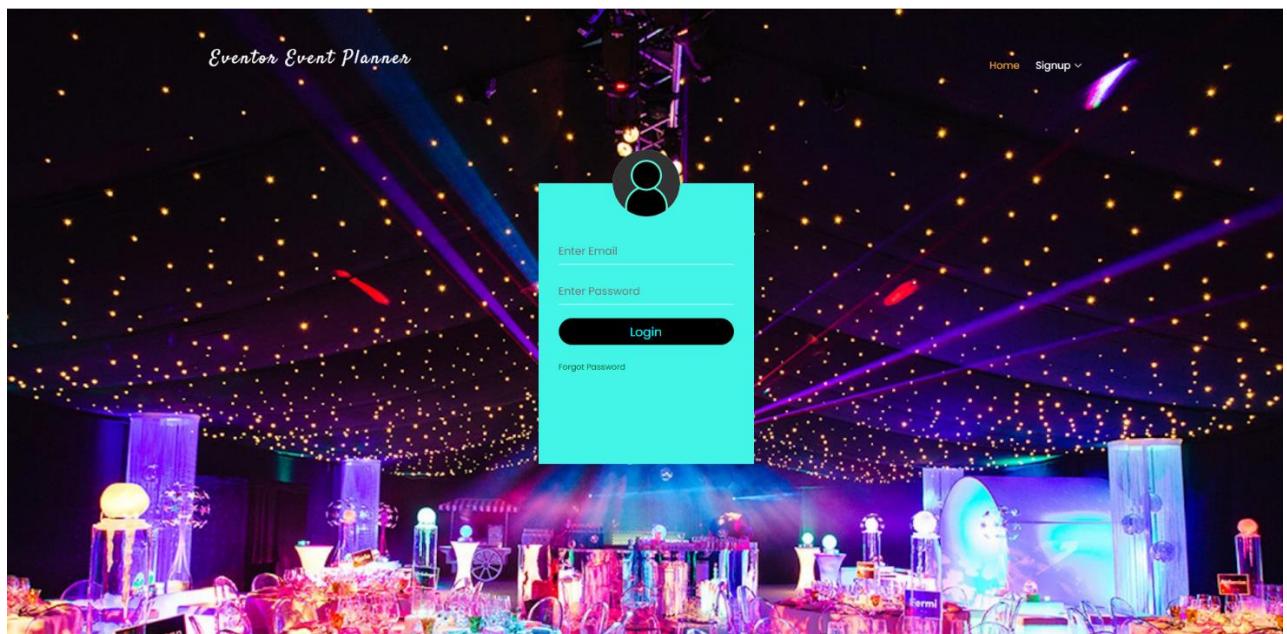


Fig 2: Login Page

USER REGISTRATION PAGE

The screenshot shows a user registration form with a black header bar containing 'Home' and 'Login' links. Below the header, a yellow arrow points down to the 'Welcome to' text and the 'User Registration Form' title. The form fields include Name, House Name, City/Landmark, Post Office, Pincode, District (a dropdown menu), Gender (radio buttons for Male, Female, Other), Phone Number, Email, Password, and Confirm Password. There is also a 'Remember me' checkbox and a note about agreeing to Terms & Privacy. At the bottom, there are 'Register' and 'Cancel' buttons.

Welcome to
User Registration Form

Please fill in this form to create an account.

Name
Enter the Name

House Name
Enter the House Name

City/Landmark
Enter the City/Landmark

Post Office
Enter Postoffice

Pincode
Enter the Pincode

District
--Please Select your District--

Gender Male Female Other

Phone Number
Enter the Phone Number

Email
Enter the EmailId

Password
Example@123

Confirm Password
Re-enter the Password

Remember me

By creating an account you agree to our [Terms & Privacy](#).

Register **Cancel**

Fig 3: User Registration Form

COMPANY REGISTRATION PAGE

The screenshot shows a registration form titled "Company Registration Form". At the top right, there are "Home" and "Login" links. Below the title, a yellow arrow points down to the "Welcome" message. The form fields include: "Company Name" (placeholder: Enter the Company Name), "Office Name/Building Name" (placeholder: Enter the Office Name), "City/Landmark" (placeholder: Enter the City/Landmark), "Post Office" (placeholder: Enter the Postoffice), "Pincode" (placeholder: Enter the Pincode), "District" (dropdown menu placeholder: --Please Select your District--), "Phone Number" (placeholder: Enter the Phone Number), "Email" (placeholder: Enter the EmailId), "Password" (placeholder: Example@123), "Confirm Password" (placeholder: Re-enter the Password), and a checked checkbox for "Remember me". A note at the bottom states: "By creating an account you agree to our [Terms & Privacy](#)". At the bottom right are "Register" and "Cancel" buttons.

Welcome

Company Registration Form

Please fill in this form to create an account.

Company Name
Enter the Company Name

Office Name/Building Name
Enter the Office Name

City/Landmark
Enter the City/Landmark

Post Office
Enter the Postoffice

Pincode
Enter the Pincode

District
--Please Select your District--

Phone Number
Enter the Phone Number

Email
Enter the EmailId

Password
Example@123

Confirm Password
Re-enter the Password

Remember me

By creating an account you agree to our [Terms & Privacy](#).

Register Cancel

Fig 4: Company Registration Form

ADMIN PAGES

Admin Home

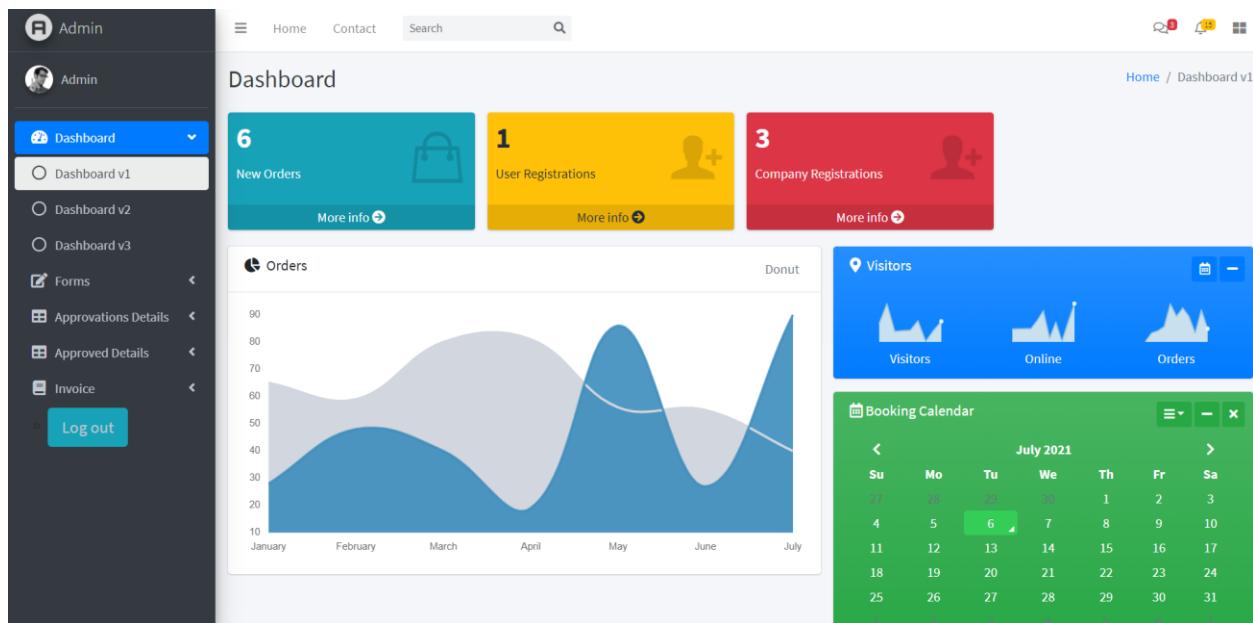


Fig 5: Admin Home

Add Food Type

The screenshot shows the "Add Food Type" page with the following sections:

- Left Sidebar:** A vertical sidebar titled "Admin" containing a user profile icon and the name "Alexander Pierce". It includes a dropdown menu for "Dashboard" (selected), "Forms", "Approvals Details", "Approved Details", "Invoice", and a "Log out" button.
- Top Header:** A navigation bar with "Home", "Contact", "Search", and a magnifying glass icon.
- Title:** "Add Food Type" with a "Home / Add Food Type" link.
- Buttons:** "ADD FOOD CATEGORY" and "Add Food Category".
- Added Food Category Details:** A table showing one entry:

Sl No	Food Category	Status	Action	EDIT	DELETE
1	Veg	Inactive	--Action--	<button>EDIT</button>	<button>DELETE</button>
- Deleted Food Category Details:** A table showing one entry:

Sl No	Food Category	Action
2	Nonveg	<button>ADD</button>
- Added Food Category Details:** A table showing one entry:

Sl No	Food Category	Status	Action
3	Veg&Nonveg	Active	--Action--

Fig 6: Food Type Add

Approve Cab Details

The screenshot shows the 'Pending Vehicle Details' section with one entry:

SL NO	Company Name	Vehicle Category	Cab Type	Taxi Cab Image	Seating Capacity	Quantity	Perday Wise Taxi Package Rate Min 8hours(Including Driver Bata)	Extra Kilometer Rate	Status	Action
3	Larza Events	Taxi	Toyota Innova A/C		7	10	2300	12.00	Pending	<button>--Action--</button>

Showing 1 to 1 of 1 entries

The screenshot also shows the 'Approved Vehicle Details' section with two entries:

SL NO	Company Name	Vehicle Category	Cab Type	Taxi Cab Image	Seating Capacity	Quantity	Perday Wise Taxi Package Rate Min 8hours(Including Driver Bata)	Extra Kilometer Rate	Status	Action
1	Megha Events	Taxi	Toyota Innova A/C		7	8	1400	12.00	Approved	<button>--Action--</button>
2	Megha Events	Premium or Classic	Jaguar XF		5	6	12000	20.00	Approved	<button>--Action--</button>

Fig 7: Approve Cab Details

Booking Analysis

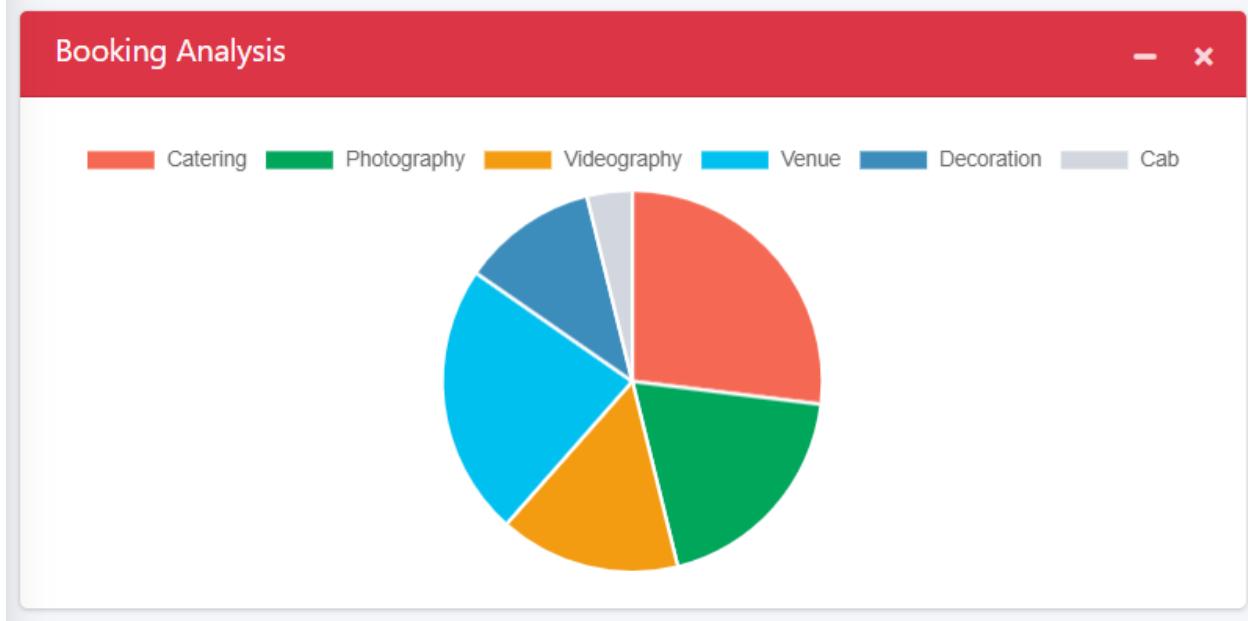


Fig 8 : Booking Analysis

COMPANY HOME

Company Home

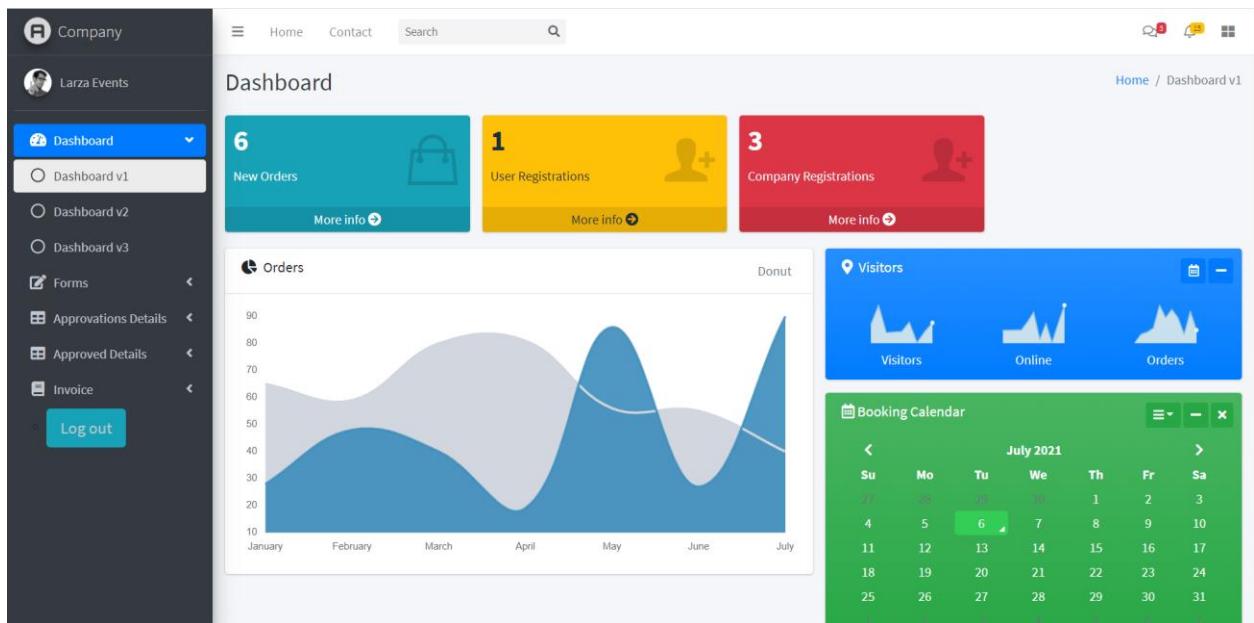


Fig 9: Company Home

Customer Order Approval

The screenshot shows the Customer Order Approval page. The sidebar includes Dashboard v1, Forms, Approved Details, and Log out. The main content is titled 'Customer Order Details' and displays a table of three orders. Each row includes columns for SL NO, Order Number, Item Name, Image, Item Quantity, Amount, Total Amount, Order Date, Booking Date, Booking Time, Status, and Action. The first two orders are marked as 'Approved', while the third is pending approval. A search bar and pagination controls are at the bottom of the table.

SL NO	Order Number	Item Name	Image	Item Quantity	Amount	Total Amount	Order Date	Booking Date	Booking Time	Status	Action
1	1	Normal Photography		3	4500	13500	14-06-2021	22-06-2021	09:30 AM	Approved	[Action]
2	2	Fortuner		1	4000	4000	18-09-2021	22-06-2021	08:40 AM	Approved	[Action]
3	3	Wedding Decoration		1	12000	12000	22-06-2021	28-06-2021	10:00 AM	Approved	[Action]

Fig 10: Customer Order Approval

USER PAGES

Booking Page

Popular Bookings

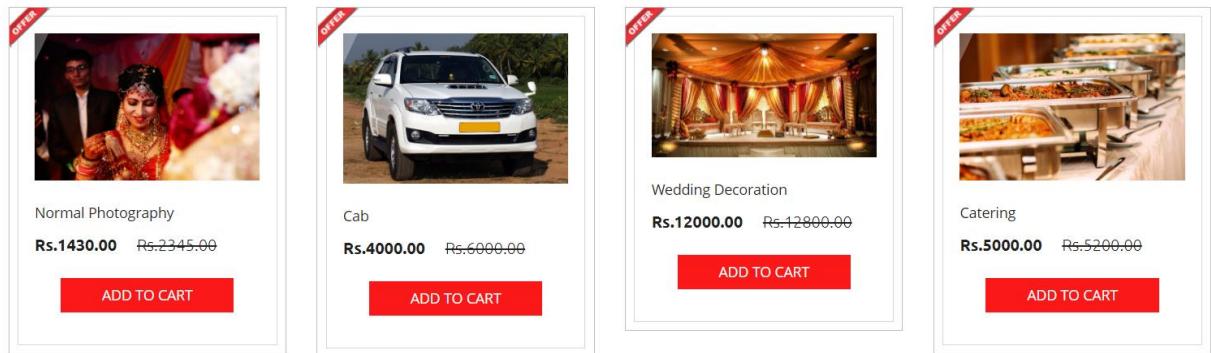


Fig 11: Popular Bookings

Single Page

NORMAL PHOTOGRAPHY

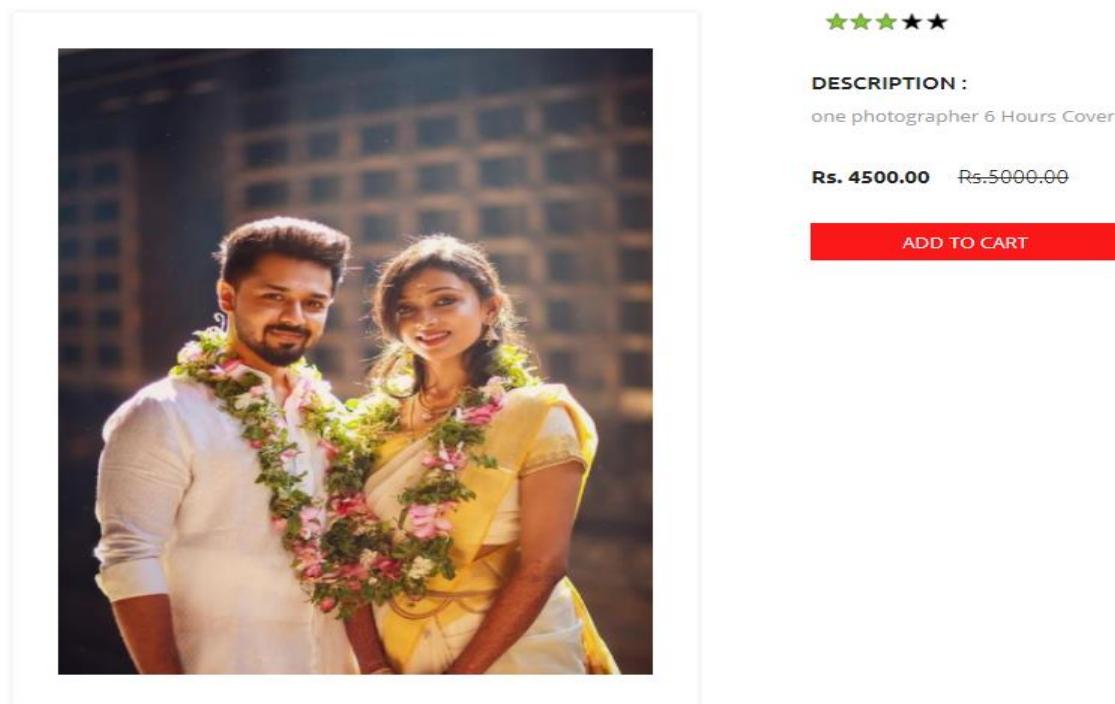


Fig 12: Single Page

Checkout Form

Checkout					
YOUR SHOPPING CART CONTAINS: ITEMS					
Sl. No.	Image	Quantity Needed	Item Name	Price	Remove
1		3	Normal Photography	Rs.13500.00	
2		1	Fortuner	Rs.4000.00	
3		1	Wedding Decoration	Rs.12000.00	

Fig 13: Checkout Page

Bill Form

CONTINUE TO BASKET		ADD A NEW DETAILS	
Item1 -	Rs. 13,500.00	Full name:	<input type="text"/>
Item2 -	Rs. 4000.00	Mobile number:	<input type="text"/>
Item3 -	Rs. 12000.00	Landmark:	<input type="text"/>
Total Service Charges -	Rs. 100.00	Town/City:	<input type="text"/>
<hr/> Total -	Rs. 29,600.00	Address type:	<input type="text"/>
DELIVERY TO THIS ADDRESS			
<input type="button" value="Make a Payment ➔"/> 			

Fig 14: Bill Form

Payment Page

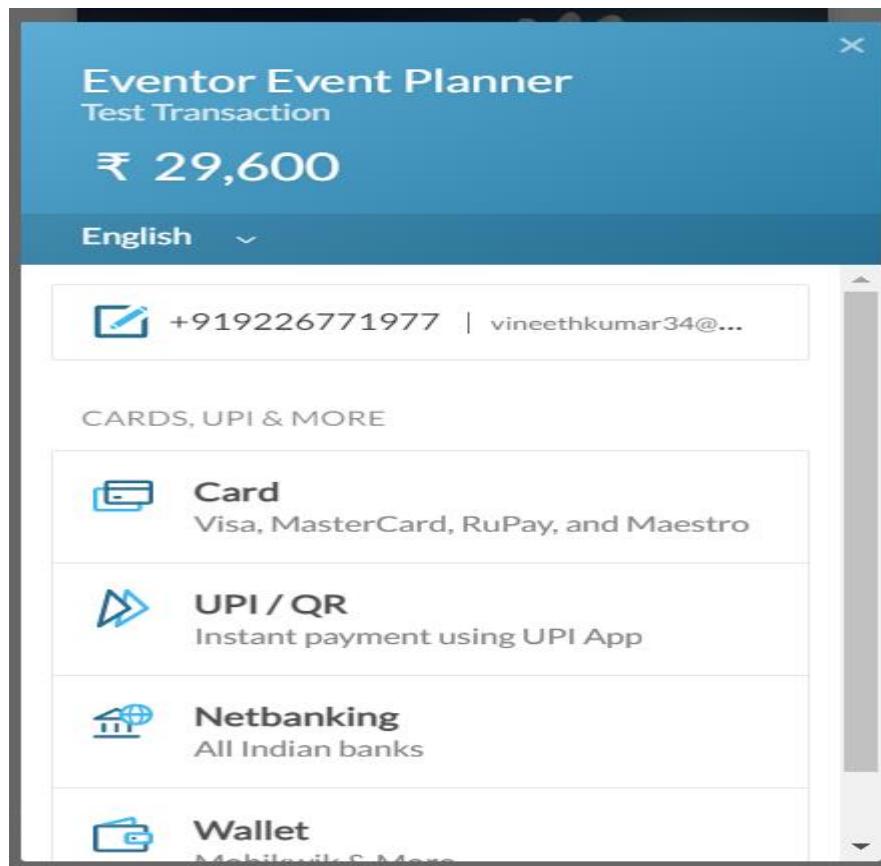


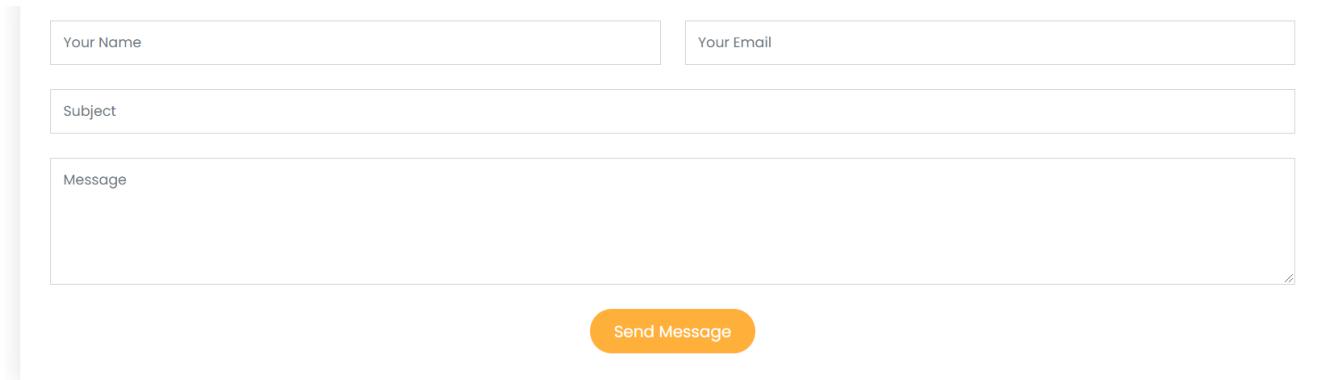
Fig 15: Payment Form

User Order History

Order Details											
Show 10 entries <input type="button" value="Search"/> Search: <input type="text"/>											
SL NO	Order Number	Item Name	Image	Item Quantity	Amount	Total Amount	Order Date	Booking Date			
1	1	Normal Photography		3	4500	13500	14-06-2021	22-06-2021			
2	2	Fortuner		1	4000	4000	18-09-2021	22-06-2021			
3	3	Wedding Decoration		1	12000	12000	22-06-2021	28-06-2021			

Showing 1 to 3 of 3 entries Previous Next

Fig 16: User Order History

Feedback Form

The image shows a feedback form with a light gray background and a white input area. At the top left is a 'Your Name' input field, followed by a 'Your Email' input field at the top right. Below these are 'Subject' and 'Message' input fields. A large orange 'Send Message' button is centered at the bottom.

Your Name

Your Email

Subject

Message

Send Message

Fig 17: Feedback Form