**FYP II Final Report**

**EVENTIFY**

|  |  |
| --- | --- |
| Project Code: | 20f-37 |
| Project Supervisor: | **Dr. Raheel Ahmed Memon** |
| Project Co-Supervisor: | **N/A** |
| Project Manager: |  |
| Project Team: | Meharwan (023-20-0123)  Kashish Kumari(023-20-0102)  Meharwan Kumar(023-20-0140) |
| Submission Date: |  |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Signature of the Project Supervisor

**Document Information**

|  |  |
| --- | --- |
| **Category** | **Information** |
| Customer | Institutes |
| Project Title | EVENTIFY |
| Document | FYP II Final Report |
| Document Version | 1.0 |
| Identifier | 20-f37 Final Report |
| Status | Final |
| Author(s) | Meharwan , Kashish Kumari , Meharwan Kumar |
| Approver(s) | Dr. Raheel Ahmed Memon |
| Issue Date |  |
| Distribution | * Advisor/Supervisor |

**Definition of Terms, Acronyms and Abbreviations**

*This section should provide the definitions of all terms, acronyms, and abbreviations required to interpret the terms used in the document properly.*

|  |  |
| --- | --- |
| **Term** | **Description** |
|  |  |
| IDE | Integrated Development Environment |
| VSC | Visual Studio Code |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# **DEDICATION**

*This report is a tribute to our cherished parents and teachers, who have served as a continual source of motivation and have provided us with financial, moral, and spiritual support when we were on the verge of giving up. Especially, to Late Prof. Nisar Ahmed Siddiqui, who provided us the opportunity to study at such an esteemed institution.*

# **ACKNOWLEDGEMENTS**

We sincerely thank the Almighty God for providing us with the knowledge we needed to plan and carry out this project. Never was there a shortage or a need. He took charge of anything that could have gotten in the way of this development and supported us through the toughest times.

We respect the Late Professor Nisar Ahmed Siddiqui, the Vice Chancellor of the great institution, because he has pursued God's guidance and creation and nurtured the vision of Sukkur IBA University, a diverse and a world-class institution where we were educated to compete and succeed in the entire world. We cannot disregard the head of the department of computer science and other faculty members who pay attention to and behave in a way that benefits students. Nevertheless, it was our teachers who encouraged and aided us in achieving our objective.

Table of Contents

[**DEDICATION 3**](#_Toc162355531)

[**ACKNOWLEDGEMENTS 4**](#_Toc162355532)

[**ABSTRACT 7**](#_Toc162355533)

[**INTRODUCTION 8**](#_Toc162355534)

[**1. Introduction 8**](#_Toc162355535)

[**1.1Project Background and problem Description 8**](#_Toc162355536)

[**1.2 Intended Audience 8**](#_Toc162355537)

[**1.3 Project Scope 9**](#_Toc162355538)

[**1.4 Not in Scope 9**](#_Toc162355539)

[**1.5 Objectives 9**](#_Toc162355540)

[**1.6 Summary 9**](#_Toc162355541)

[**LITERATURE REVIEW 11**](#_Toc162355542)

[**REQUIREMENTS 13**](#_Toc162355543)

[**3. Functional Requirements 13**](#_Toc162355544)

[**3.1 Functional Hierarchy 13**](#_Toc162355545)

[**3.2 Use Cases 14**](#_Toc162355546)

[**3.3 Non-Functional Requirement 26**](#_Toc162355547)

[**3.3.1 Performance Requirement 26**](#_Toc162355548)

[**3.3.2 Safety Requirements 26**](#_Toc162355549)

[**3.3.3 Security Requirements 26**](#_Toc162355550)

[**PROPOSED APPROACH 28**](#_Toc162355551)

[**4. Methodology 28**](#_Toc162355552)

[**4.1 Agile Methodology 28**](#_Toc162355553)

[**4.2 System Architecture 28**](#_Toc162355554)

[**4.3 Operating Environment 30**](#_Toc162355555)

[**4.4 System Restrictions 30**](#_Toc162355556)

[**4.5 Design 30**](#_Toc162355557)

[**4.5.1 Class Diagram 30**](#_Toc162355558)

[**4.5.2 Entity Relationship Diagram (ERD) 31**](#_Toc162355559)

[**4.5.3 Sequence Diagrams 32**](#_Toc162355560)

[**IMPLEMENTATION 35**](#_Toc162355561)

[**5.1 Overview 35**](#_Toc162355562)

[**5.2 Implementation 35**](#_Toc162355563)

[**5.3 Front End Using React JS 35**](#_Toc162355564)

[**5.4 Visual Studio Code 35**](#_Toc162355565)

[**5.5 Backend (Node.JS) 36**](#_Toc162355566)

[**5.6 MongoDB 36**](#_Toc162355567)

[**TESTING 39**](#_Toc162355568)

[**6.1 Overview 39**](#_Toc162355569)

[**6.2 Test Classifications 39**](#_Toc162355570)

[**6.3 Test Cases 40**](#_Toc162355571)

# 

# **ABSTRACT**

In today's technological age, everyone has everything planned out. They aim to accomplish things in a way that is less expensive and more effective. However, meeting preparations, wedding decorations, or any other type of event can be done with a single click.

Our final year project concept is to build a platform on which people may trust. Which supplies them with the greatest areas or locations in which to hold their events or meetings. Our main goal will be to satisfy clients and meet their demands within their budget.

Eventify, an innovative online platform, redefines event planning by providing an organized solution for a wide range of event requirements. Eventify reduces the planning process for both urban and rural areas of Pakistan by offering a varied range of event venues and services ranging from wedding halls to conference rooms. The platform promotes smooth connections between clients and service providers by using customized vendor profiles and secure payment mechanisms. Eventify wants to make event planning an easy, stress-free, and memorable experience, redefining how events and celebrations are planned.

***CHAPTER 1***

# **INTRODUCTION**

## **1. Introduction**

In an ever-evolving world where celebrations and gatherings hold profound significance, the art of event planning has assumed a pivotal role. From intimate family reunions to grand corporate conferences, the orchestration of events demands meticulous coordination, creativity, and seamless execution. However, the traditional approach to event planning often involves a maze of difficulties, from finding appropriate venues to hiring reliable service providers, all while ensuring that guests have a great experience.

A paradigm change is taking place against this background. Introducing "Eventify," a revolutionary web platform which will alter the way events are planned. As a complete solution, Eventify is ready to automate the entire event planning procedure and bring in a new era of comfort, accessibility, and improved experiences.

The difficulties experienced along the way have served as a spark for creativity, leading to the development of a platform that not only removes obstacles but also pioneers a fresh perspective on celebrations. The foundation of Eventify's promise to revolutionize the event planning industry and provide a platform where innovation seamlessly meets celebration.

Let's explore the detailed variety that is Eventify as we set out on this journey of discovery. Let's look at the features, the methodology, the aspirations, and the shared dedication to creating a future where event planning is characterized by simplicity, effectiveness, and limitless potential. Welcome to the world of Eventify, where each event presents a unique opportunity to create an unforgettable story.

## **Project Background and problem Description**

The traditional approach to event planning frequently poses a variety of obstacles in today's changing world of celebrations and gatherings, ranging from time-consuming logistics to fragmented solutions. Whether it's a professional or personal event, organizing involves sifting through a jumble of responsibilities such as choosing a location, coordinating vendors, and handling guests. Furthermore, the pandemic's start has made it even more urgent to find streamlined, digital solutions that put efficiency and safety first. Identifying these problems and possibilities, Eventify surfaces as a revolutionary platform ready to completely change the event organizing sector. To address the intricacies and inefficiencies present in conventional approaches, Eventify attempts to automate and streamline the whole process. Eventify aims to facilitate accessibility for users of different backgrounds and does away with the necessity for numerous tools.

## **Intended Audience**

The target market for Eventify consists of people and businesses from a range of industries who manage and organize events. Managers, organizers, and event planners who serve in both personal and professional roles fall under this category. Furthermore, Eventify caters to companies and organizations looking for effective ways to plan conferences, seminars, corporate events, and product launches. Additionally, Eventify helps people organize social events like weddings, anniversaries, birthday parties, and family reunions. In addition, Eventify reaches out to suppliers, vendors, service providers, and owners of venues that want to expand their customer base and simplify their services. Both beginners and seasoned pros in the event planning sector find Eventify appealing due to its user-friendly UI and extensive feature set.

## **Project Scope**

The Primary objective to create, develop, and introduce a complete event management platform that enables the smooth planning, coordination, and execution of various events from beginning to conclusion.

## **1.4 Not in Scope**

Initially, Our Main focus will be on metropolitan cities Karachi, Islamabad, Sukkur, Hyderabad and Quetta after Some time will provide our services in all over Pakistan.

## **1.5 Objectives**

Eventify’s main objective to provide a complete digital platform that helps event planning for a wide range of events. This platform will act as a hub for users to plan, organize and carry out their events, which might range from small family gathering to huge business conferences. The Primary objective to create, develop, and introduce a complete event management platform that enables the smooth planning, coordination, and execution of various events from beginning to conclusion. By assuring open communication, effective resource management, and the execution of memorable events that are in line with client visions and market trends, the platform seeks to improve the client experience. Our goal is to establish ourselves as the top choice for clients, ranging from corporate organizations to individual celebrants, by utilizing cutting-edge technologies and best practices. We do this by ensuring consistently high-quality service, adaptability to a range of event sizes and types, and the development of long-lasting client relationships.

Eventify seeks to cater to a wide range of events types, including but not limited to wedding, corporate meeting, birthday parties, family reunion and more because of its inclusion the platform is versatile and adaptable to a wide range of events.

## **1.6 Summary**

This chapter discusses the project's introduction and the features that the project is offering. The project objectives and the literature review are covered in chapter 2. Problem definition and methodology are addressed in chapters 3 and 4. The project's architecture and design are thoroughly described in chapter 5. Implementation and testing have been dealt with in chapter 6. Results and discussion are presented in chapter 7. The conclusion and future work are mentioned in chapter 8, while references are provided in chapter 9

***CHAPTER 2***

# **LITERATURE REVIEW**

The traditional approach to event planning typically involves manual coordination, limited visibility, fragmented communication, and a lack of automation. Event organizers face challenges such as manually coordinating venue selection, and vendor management leading to inefficiencies and errors.

Eventify addresses these challenges by automating workflows, centralizing collaboration, and providing customizable templates. By automating various aspects of event planning and offering a centralized platform for real-time collaboration, Eventify streamlines tasks and enhances efficiency. Integrated communication tools further improve organization and coordination, ensuring successful event execution. From intimate gatherings to large-scale conferences, Eventify revolutionizes the event planning process, empowering organizers to plan and execute events seamlessly.

To support the proposal title, we conducted extensive research from various credible sources such as academic databases, industry reports, and online platforms. In the initial stages, we gathered keywords related to event planning and formulated appropriate queries for these sources. After sifting through numerous articles based on title and abstract proofreading, we included three key sources in our analysis.

Two of the sources discussed online platforms similar to Eventify, highlighting the significant role they play in connecting event planners with service providers. These platforms offer a wide range of specialized services and facilitate communication between clients and vendors. They act as intermediaries, providing a direct connection between those in need of event services and those offering them.

Such platforms have proven to be indispensable for businesses and individuals alike, streamlining the event planning process and ensuring seamless execution. The third source focused on the benefits of utilizing online platforms for event planning. It highlighted how these platforms improve efficiency, accessibility, and overall user experience. By leveraging technology, event planners can overcome traditional challenges and create memorable experiences for their clients. Drawing inspiration from these findings, we have conceptualized Eventify as a revolutionary web platform that aims to transform the event planning industry. Eventify will automate and streamline the entire event planning process, offering a comprehensive solution for individuals and businesses seeking to organize successful events. By combining **innovation with celebration, Eventify promises to redefine the standards of event management and create unforgettable experiences for all involved.**

***We have gone through some of the Event Planning platforms in Pakistan such as:***

* Evento.com.pk
* Dawatpk.com

And we discovered that both platforms primarily concentrate wedding events while ignoring other occasions including catering, corporate meetings, and birthday parties. If they are offering additional offerings but are not paying close attention to those events. Whereas other event organizers demand excessive fees for single events, we offer packages of events with lower pricing. Our website focuses on all the events and pays equal attention to each one at favorable and reasonable prices.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Available** | Wedding | Corporate | Birthday | Catering | Combine | Suitable |
| **Websites** | Events | Conference | Parties | Service | Packages | Cost |
| Evento.com.pk |  | ***x*** | ***x*** | ***x*** | ***x*** | ***X*** |
| Dawatpk.com |  | ***x*** |  |  | ***x*** | ***X*** |
| Proposed System |  |  |  |  |  |  |

Research indicates a palpable shift among event planners towards web-based platforms, which offer a plethora of benefits such as streamlined workflow, enhanced collaboration, and access to a vast array of services. These digital hubs serve as command centers for managing every aspect of event logistics, be it venue selection, budget management, guest coordination, or marketing endeavors. By harnessing the power of data analytics and automation, event organizers are empowered to make informed decisions and optimize their strategies for maximum impact.

Eventify – a beacon of innovation and a paragon of versatility in the realm of event planning. By melding cutting-edge technology with avant-garde features, Eventify emboldens event organizers to craft immersive and captivating experiences that captivate audiences and etch indelible memories. In essence, Eventify isn't just a platform; it's a catalyst for transformation, heralding a new era where events are not merely organized but orchestrated with finesse and flair.

In essence, as we delve deeper into the intricate tapestry of event planning, it becomes abundantly clear that Eventify isn't just a solution; it's a revolution – a harbinger of change that promises to redefine the very essence of how we conceive, execute, and savor the moments that matter most. Welcome to the world of Eventify, where every event presents an opportunity to write a unique and unforgettable story.

***CHAPTER 3***

# **REQUIREMENTS**

## **3. Functional Requirements**

* **User-friendly interface:** Eventify boasts an intuitive and easy-to-navigate interface designed to simplify the event planning process. With clear layouts, intuitive controls, and visually appealing design elements, users can seamlessly navigate the platform and access its features without any hassle.
* **Seamless integration:** Eventify offers seamless integration with various third-party tools and services, allowing users to leverage their existing resources and workflows. Whether it's integrating with popular social media platforms for event promotion or syncing with calendar apps for scheduling, Eventify ensures a smooth and cohesive user experience.
* **Real-time collaboration:** Eventify facilitates real-time collaboration among event organizers, vendors, and participants, enabling them to work together efficiently and effectively. With features such as shared calendars, collaborative document editing, and instant messaging, teams can coordinate tasks, share updates, and stay in sync throughout the event planning process.
* **Customizable features:** Eventify provides a range of customizable features that cater to the unique needs and preferences of each event. From customizable event templates and branding options to flexible scheduling tools and personalized communication channels, Eventify empowers users to tailor their event planning experience to suit their specific requirements.

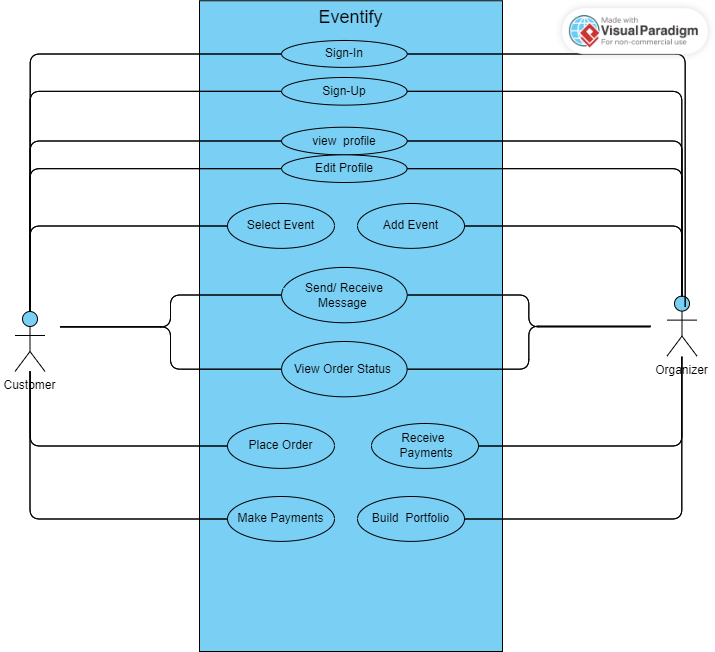
## **3.1 Functional Hierarchy**

This section will give a big picture of overall system functionality.

* A sign-up page for new clients must be provided by the system.
* Existing clients must be able to sign in to use the system.
* When users click the "I forgot my password" link, the system must allow them to reset their password.
* When users register as organizer, the system should ask for information such as experience, portfolio, and so on.
* The system must provide its users with a homepage.
* The system must enable customers to select their package.
* Organizers must be able to access, modify, and set their profile/portfolio in the system.
* The system should allow customers to search for best package according to their need.
* The system should allow customers to compare prices of different packages.

The system should provide a page where organizers can view completed and pending orders.

## **3.2 Use Cases**

**

*Figure 3.2: Use Cases*

Use Case Description

|  |  |  |  |
| --- | --- | --- | --- |
| **01: Sign Up** | | | |
| **Use case Id:** | | 01 | |
| **Actors:** | | Organizer, Customer | |
| **Pre-condition:** | | The user has a correct email address. | |
| **Scenarios** | | | |
| **Step #** | **Action** | **Software Reaction** | |
| **1.** | Enter email address | The provided email address will be validated whether it is written in correct pattern or not. | |
| **2.** | Enter password | The provided password will be checked whether it is of certain length or not i.e., 8 or more characters. | |
| **3.** | Click “Sign Up” button | The email address and password will be used create a new account if it does not already exist. | |
| **Alternate Scenarios** | | | |
| **1a:** If the email address entered is incorrect, the user will be prompted.  **2a:** If the password contains less than 8 characters, the user will be prompted.  **3a:** If an account already exists against the entered email address, then the user will be prompted. | | | |
| **Post Conditions** | | | |
| **Step #** | **Description** | | |
| **1.** | When the user fills in correct information in the sign-up form, an account will be created against his/her email address. | | |
| **Use Case Cross referenced** | | | **Use Cases** |

|  |  |  |  |
| --- | --- | --- | --- |
| **02: Sign In** | | | |
| **Use case Id:** | | 02 | |
| **Actors:** | | Organizer, Customer | |
| **Pre-condition:** | | The user has an existing account on the platform. | |
| **Scenarios** | | | |
| **Step #** | **Action** | **Software Reaction** | |
| **1.** | Enter email address | The provided email address will be validated whether it is written in correct pattern or not. | |
| **2.** | Enter password | The provided password will be checked whether it is of certain length or not i.e., 8 or more characters. | |
| **3.** | Click “Sign In” button | The email address and password will be used to check whether the account exists or not. | |
| **Alternate Scenarios** | | | |
| **1a:** If the email address entered is incorrect, the user will be prompted.  **2a:** If the password entered is incorrect, the user will be prompted.  **3a:** If an account does not exist against the entered email address, then the user will be prompted. | | | |
| **Post Conditions** | | | |
| **Step #** | **Description** | | |
| **1.** | When the user fills in correct information in the sign-in form, he/she will be directed towards the homepage and will be able to access the platform. | | |
| **Use Case Cross referenced** | | | **Use Cases** |

|  |  |  |  |
| --- | --- | --- | --- |
| **03: View Profile** | | | |
| **Use case Id:** | | 03 | |
| **Actors:** | | Organizer, Customer | |
| **Pre-condition:** | | The user is signed-in into the platform. | |
| **Scenarios** | | | |
| **Step #** | **Action** | **Software Reaction** | |
| **1.** | Click “Profile” | The user will be directed towards his/her profile. | |
| **Alternate Scenarios** | | | |
| Not applicable | | | |
| **Post Conditions** | | | |
| **Step #** | **Description** | | |
| **1.** | The user will be able to view his/her details on the profile. | | |
| **Use Case Cross referenced** | | | **Use Cases** |

|  |  |  |  |
| --- | --- | --- | --- |
| **04: Edit Profile** | | | |
| **Use case Id:** | | 04 | |
| **Actors:** | | Organizer, Customer | |
| **Pre-condition:** | | The user is signed-in into the platform. | |
| **Scenarios** | | | |
| **Step #** | **Action** | **Software Reaction** | |
| **1.** | Click “Profile” | The user will be directed towards his/her profile. | |
| **2.** | Click “Edit Profile” | The user will be able edit the details of his/her profile i.e., Profile Picture, Contact Details, etc. | |
| **3.** | Save or cancel | The changes will be saved or ignored. | |
| **Alternate Scenarios** | | | |
| Not applicable | | | |
| **Post Conditions** | | | |
| **Step #** | **Description** | | |
| **1.** | After saving the changes, the user will be able to view the changed information on his/her profile. | | |
| **Use Case Cross referenced** | | | **Use Cases** |

|  |  |  |  |
| --- | --- | --- | --- |
| **05: Select Event** | | | |
| **Use case Id:** | | 05 | |
| **Actors:** | | Customer | |
| **Pre-condition:** | | The user is signed-in into the platform. | |
| **Scenarios** | | | |
| **Step #** | **Action** | **Software Reaction** | |
| **1.** | Click “We Offer” | The user will be directed towards the “events” section. | |
| **2.** | Select a suitable package | The user can provide title, description, and time duration, ask price, etc. for the package. | |
| **Alternate Scenarios** | | | |
| Not applicable | | | |
| **Post Conditions** | | | |
| **Step #** | **Description** | | |
| **1.** | When the package is selected, it will be visible in the “View Orders” section. | | |
| **Use Case Cross referenced** | | | **Use Cases** |

|  |  |  |  |
| --- | --- | --- | --- |
| **06: Add Event** | | | |
| **Use case Id:** | | 06 | |
| **Actors:** | | Organizer | |
| **Pre-condition:** | | The user is signed-in into the platform. | |
| **Scenarios** | | | |
| **Step #** | **Action** | **Software Reaction** | |
| **1.** | Click “We Offer” | The user will be directed towards the “events” section. | |
| **2.** | Select an event | The user can select any job from the list and can provide, description, time duration, ask price, etc. for that event. | |
| **3.** | Click “Bid” | The offer will be visible on the client web screen. | |
| **Alternate Scenarios** | | | |
| Not applicable | | | |
| **Post Conditions** | | | |
| **Step #** | **Description** | | |
| **1.** | The offer will be successfully sent to the client upon providing all the required information. | | |
| **Use Case Cross referenced** | | | **Use Cases** |

|  |  |  |  |
| --- | --- | --- | --- |
| **07: View Order Status** | | | |
| **Use case Id:** | | 07 | |
| **Actors:** | | Organizer, Customer | |
| **Pre-condition:** | | The user is signed-in into the platform. | |
| **Scenarios** | | | |
| **Step #** | **Action** | **Software Reaction** | |
| **1.** | Click “Orders” | The user will be directed to the “Orders” page. | |
| **2.** | View order details | From the list of orders, user can select any order to view details such as response, remaining time, price, requirements, etc. | |
| **Alternate Scenarios** | | | |
| Not applicable | | | |
| **Post Conditions** | | | |
| **Step #** | **Description** | | |
| **1.** | Order details will be displayed individually for each order. | | |
| **Use Case Cross referenced** | | | **Use Cases** |

|  |  |  |  |
| --- | --- | --- | --- |
| **08: Place Order** | | | |
| **Use case Id:** | | 08 | |
| **Actors:** | | Customer | |
| **Pre-condition:** | | The user is signed-in into the platform. | |
| **Scenarios** | | | |
| **Step #** | **Action** | **Software Reaction** | |
| **1.** | Accept Organizer’s offer | The user can place an order by accepting any Organizer’s offer. | |
| **2.** | Send custom order | The user can place an order by sending a custom order by visiting his/her profile. | |
| **Alternate Scenarios** | | | |
| Not applicable | | | |
| **Post Conditions** | | | |
| **Step #** | **Description** | | |
| **1.** | After placing an order, it can be viewed in the “Orders” section. | | |
| **Use Case Cross referenced** | | | **Use Cases** |

|  |  |  |  |
| --- | --- | --- | --- |
| **09: Make Payments** | | | |
| **Use case Id:** | | 09 | |
| **Actors:** | | Customer | |
| **Pre-condition:** | | The user is signed-in into the platform. | |
| **Scenarios** | | | |
| **Step #** | **Action** | **Software Reaction** | |
| **1.** | Making payment | After accepting or sending order, the user will have to make a payment for the order which can be achieved through various methods. | |
| **Alternate Scenarios** | | | |
| **1a:** In case of unsuccessful payment, the user will be prompted. | | | |
| **Post Conditions** | | | |
| **Step #** | **Description** | | |
| **1.** | When the payment is done, it will be successfully sent in escrow. | | |
| **Use Case Cross referenced** | | | **Use Cases** |

|  |  |  |  |
| --- | --- | --- | --- |
| **10: Receive Payments** | | | |
| **Use case Id:** | | 10 | |
| **Actors:** | | Organizer | |
| **Pre-condition:** | | The user is signed-in into the platform. | |
| **Scenarios** | | | |
| **Step #** | **Action** | **Software Reaction** | |
| **1.** | Receiving payment | When the order is completed, the organizer will receive the payment into his/her account. | |
| **Alternate Scenarios** | | | |
| Not applicable | | | |
| **Post Conditions** | | | |
| **Step #** | **Description** | | |
| **1.** | After successfully completion of the order, the payment will be released from escrow to the organizer’s account. | | |
| **Use Case Cross referenced** | | | **Use Cases** |

|  |  |  |  |
| --- | --- | --- | --- |
| **11: Build Portfolio** | | | |
| **Use case Id:** | | 11 | |
| **Actors:** | | Organizer | |
| **Pre-condition:** | | The user is signed-in into the platform. | |
| **Scenarios** | | | |
| **Step #** | **Action** | **Software Reaction** | |
| **1.** | Click “Profile” | The user will be directed to the “Profile” page. | |
| **2.** | Add projects | Organizer can add previous experience of events to showcase their talent in order to attract clients. | |
| **Alternate Scenarios** | | | |
| Not applicable | | | |
| **Post Conditions** | | | |
| **Step #** | **Description** | | |
| **1.** | Organizers’ portfolio will be displayed on their profiles. | | |
| **Use Case Cross referenced** | | | **Use Cases** |

## **3.3 Non-Functional Requirement**

This section outlines the fundamental non-functional requirements, which are also called the quality attributes of the system under discussion

### **3.3.1 Performance Requirement**

Performance is the measure of a system's response time. The system's performance is correlated with the following requirements:

* The web browser must function well on the system. It needs to be made to load in as little as 1 to 5 seconds, depending on the user's internet speed.
* For a web-based event management system to manage heavy traffic and offer an impeccable user experience, it must be quick, dependable, secure, and scalable. Along with ensuring browser compatibility, content optimization, disaster recovery, and mobile device friendliness, it should also have these features. It's also critical to keep an eye on performance
* User responses from the database and server operations must be prompt.

### **3.3.2 Safety Requirements**

The requirements related to the extent to which the system does not endanger life or property are known as safety requirements. A few safety requirements are as follows:

* When in use, the system must not crash.
* Regularly creating copies of your data and having a plan in place for retrieving it in the event of an emergency, such as a system failure or data loss, are known as data recovery and backup.
* Defense against SQL Injection and Avoidance of Cross-Site Scripting (XSS).

### **3.3.3 Security Requirements**

Security requirements outline how well the system guards against unwanted use, access, and modification. Here are some prerequisites for security:

* Ensure that only authorized users are able to access the system and carry out their assigned tasks.
* Encrypt confidential data so that only authorized individuals can decipher it.
* Install robust defenses to keep rogue actors out and stop them from causing havoc. This includes preventing hackers from accessing the system or stealing data.

***CHAPTER 4***

# **PROPOSED APPROACH**

## **4. Methodology**

We used the agile technique, which divides a project's management into distinct phases, to construct this project. The following four values are stated in the Manifesto for Agile Software Development.

* People and their interactions with procedures and equipment.
* Functional software as opposed to thorough documentation.
* Customer participation during contract negotiations.
* Adapting to change rather than adhering to a plan.

This paradigm allows for flexibility in project development, which enabled us to accommodate modifications as they arose.

## **4.1 Agile Methodology**

Although agile software development approaches were initially introduced in the 1990s, their significant rise in popularity began in 2001. They also stress the need of agility in software development. The capacity to change quickly and efficiently is known as agility. The easiest way for developing software is the agile development process since it is team-based and iterative. Agile development helps us achieve our goals by using sprints as an alternative to timelines and deadlines.  
We have noticed the following benefits of agile development during our development cycle:

* It was very easy to use and straightforward.
* We were able to stay focused throughout the whole project's production.
* We are free to make changes as needed while our app is being built.
* The final result was of the greatest quality.
* The method for overseeing the project's progress was more efficient.

## **4.2 System Architecture**

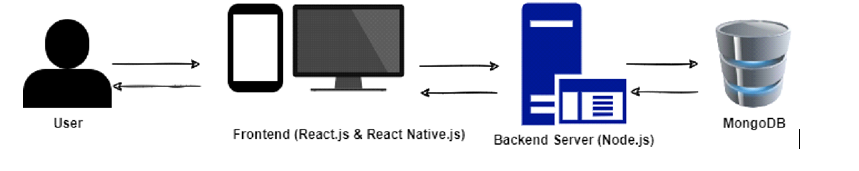
A three-tier architecture is the foundation of the project. It is a popular and extensively employed architectural style. The application is divided into three levels by it. The front-end, or user interface, is the initial layer and is where users will interact with the system. The business logic is housed in the second tier, which is server-side, while data is stored in the third layer, which is called the data storage tier [5]. This design has the advantage of breaking up the application into three distinct levels, allowing each to be created, scaled, and updated independently by various team members without affecting the other layers. The provided three-tier design of the web application and the mobile application is shown in operation in Figure 4.2

Three Tier Architecture consists of the following modules:

**Presentation tier, or front-end**: We used React JS for the proposed application for front-end (client-side) development. In MERN, React JS is on the top of the stack. React is a declarative framework by which dynamic client-side applications can be built in HTML. Moreover, react allows the building complex interfaces with simple components.

**Server or Back-end**: Using the Node JS framework for the server-side or back end and Express JS for the front end, we partially constructed the suggested system. With the Node JS runtime environment, programmers may create any JavaScript server-side application. An open-source, cross-platform environment was provided by Nodes JS. From a web server development standpoint, Node JS provided a number of advantages. The most widely used Node framework on the web is Express JS. Express JS is a Node JS web framework that is quick, simple, and adaptable, offering powerful functionalities for both online and mobile apps.

**Database:** MongoDB was utilized in this application to manage data in a database. A NoSQL document database is called MongoDB. NoSQL is an acronym for nontabular database, meaning "not only SQL." Structured Query Language, or SQL for short, is a relational database that stores data in tables. MongoDB is used to store JSON documents. The relational database's counterpart of tables, collections, are employed. There may be several JSON documents in these collections. In SQL, a document is equivalent to a record or row. In a SQL table, fields and columns are comparable. Mongoose served as a translator between code objects and MongoDB objects as well as managing the link between data and schema validation.



*Figure 4.2: Three-tier architecture.*

## **4.3 Operating Environment**

The system is based on web application so for using it, the user must have a mobile device, a tablet or a laptop/computer with internet connectivity, and a web browser that supports HTML5 and JavaScript.

## **4.4 System Restrictions**

* The system places the following limitations on its usage in order to prevent misuse of the services it offers and to ensure that it operates as intended:   
  **Software limitations:** As traffic volume rises, the system may experience glitches and crashes.
* **Hardware requirements:** In order to access the system, a laptop, computer, or mobile device with internet connectivity is needed.
* **Cultural restrictions:** Initially, the system is only available in English.
* **Environmental restrictions**: For the system to operate correctly, there must be active internet access.

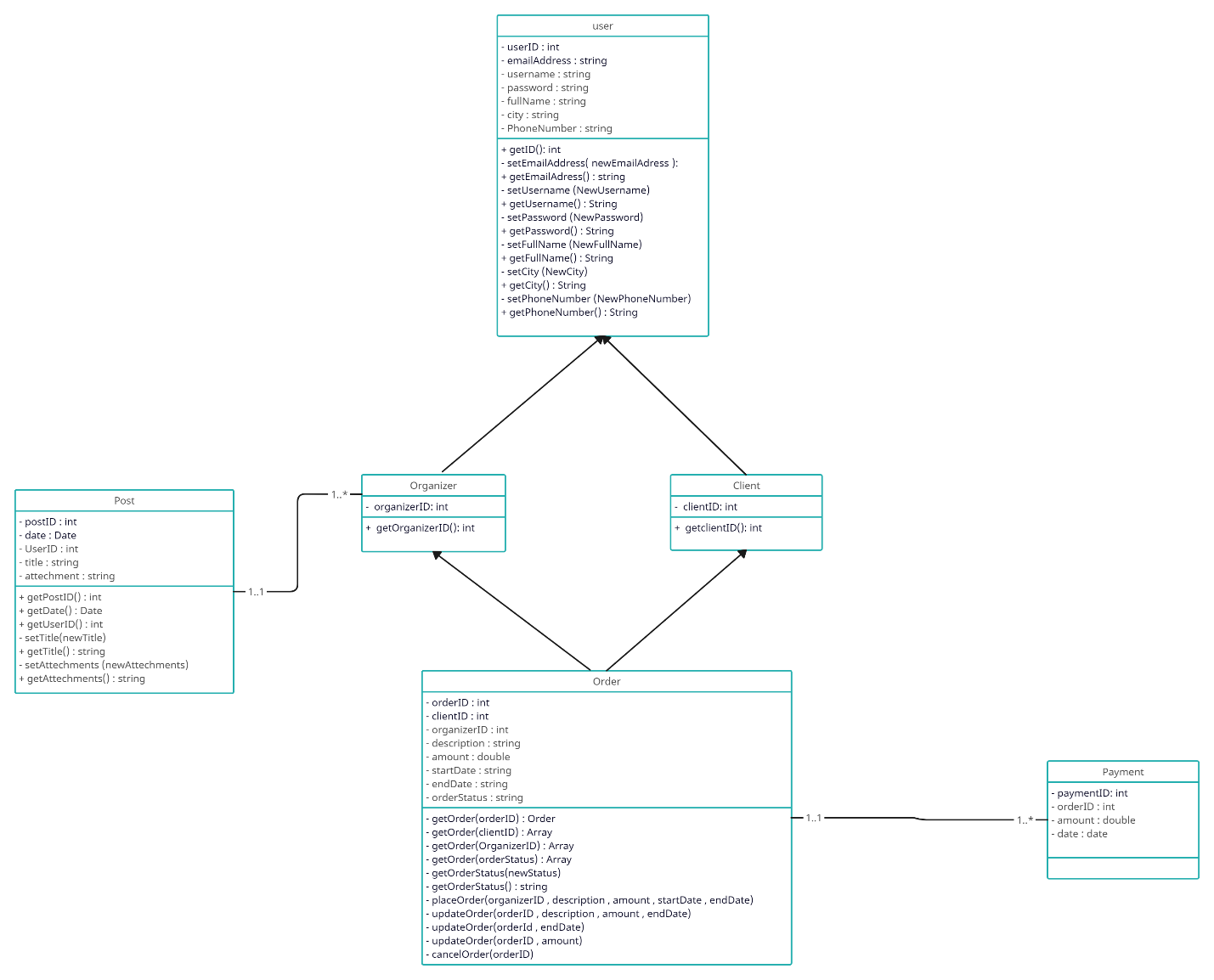
**User restrictions:** Users must be literate enough to register on the website and post jobs only in English. To work with customers, people who join as freelancers are required to create or give a portfolio of their work, which can take the shape of a website, LinkedIn page, or any other platform that allows them to display their abilities.

## **4.5 Design**

This project is designed in UML (Unified Modeling Language) using Star UML such as Sequence diagrams, class diagrams, and use case diagrams.

### **4.5.1 Class Diagram**

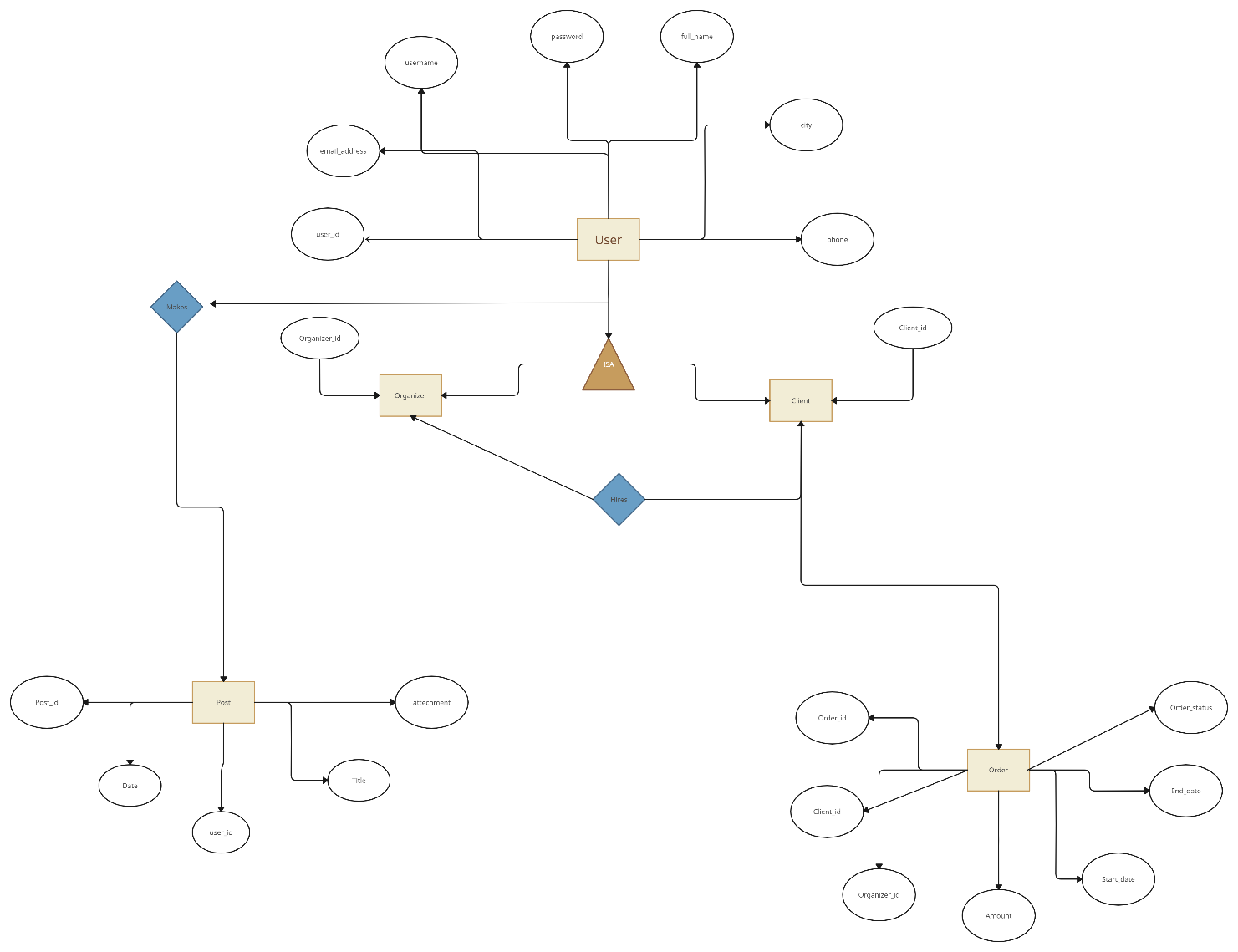
The schematics for the system or subsystem are shown in the class diagrams. Class diagrams may be used to show the relationships between the system's component parts, as well as the different functions and duties that they each perform. Class diagrams are useful in all phases of the system design process.   
Class diagrams may make you feel more comfortable. They provide you a thorough rundown of how your systems are configured. They also provide a succinct description of the characteristics and connections between the system's parts, as well as how they interact.   
The class diagram for clearing and no dues Use is displayed in Figure 4.3.1 . This diagram also defines all the classes related to our system and their relationships.



*Figure4.3.1: Class Diagram*

### **4.5.2 Entity Relationship Diagram (ERD)**

An ER diagram is used in the creation of database data models. When developing a new data model, the ER Diagram helps with the important process of doing due diligence to determine and document system requirements. During production, an ER diagram is used as a guide to improve database throughput, troubleshoot problems, and rethink architecture. The entity relationship diagram of no dues and clearance, which shows all related entities and their connections inside the system, is depicted in Figure 4.3.2.

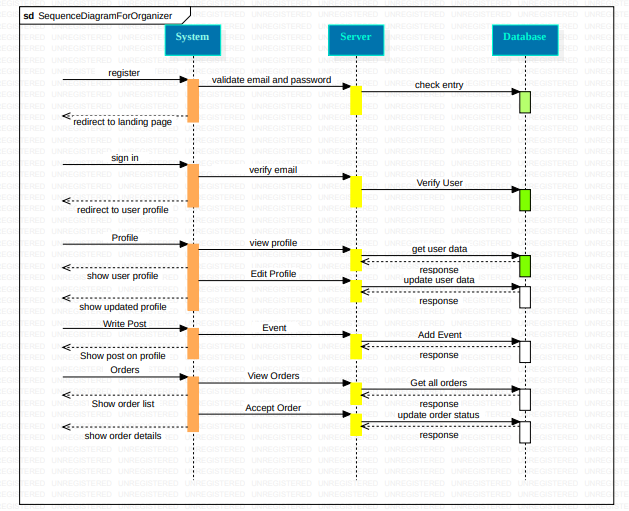


*Figure 4.3.2: Entity Relationship Diagram*

### **4.5.3 Sequence Diagrams**

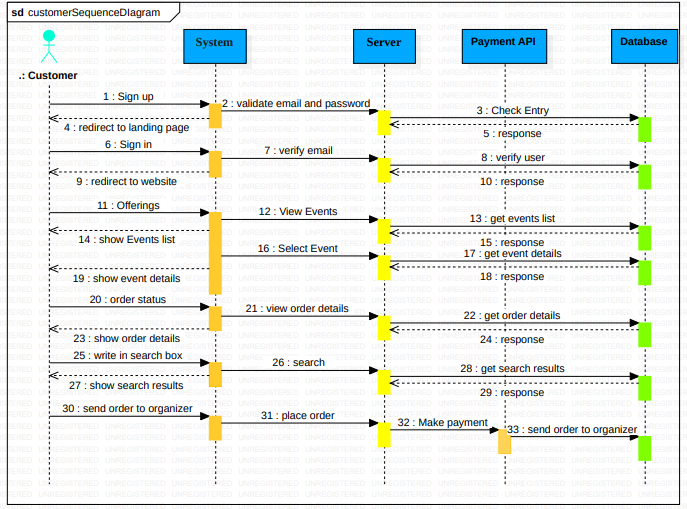
Sequence diagrams explain how procedures are carried out in fairly depth. They capture the exchange of information between objects. With a temporal focus, sequence diagrams may graphically depict the sequence of events by representing time and the messages that are transmitted at different times along the vertical axis of the diagram. The following numbers explain the actors, who include students and faculty personnel, along with their individual positions.

**Sequence Diagram for Organizer**



***Figure 4.3.3.2: Sequence Diagram for organizer***

**Sequence Diagram for Customer**



*Figure 4.3.3.3: Sequence Diagram for Customer*

***CHAPTER 5***

# **IMPLEMENTATION**

## **5.1 Overview**

A Web project's implementation. To support web applications, backend tools such as databases and APIs are used

## **5.2 Implementation**

The initial step in the implementation process was downloading and installing the required development tools and packages, such as MySQL, VSC, ES6 (ECMAScript 6), JavaScript, Node.js, Minimal libraries, and React.js.   
As indicated by the ER Diagram, database entities are created following the installation of all tools and packages. Now that MySQL has all the entities, you can begin writing code by creating the login page first.   
The Minimal template was used to develop the project's user interface. The React library is used in combination with HTML, CSS, and JavaScript to generate the system's interfaces.

## **5.3 Front End Using React JS**

React is a component-based framework for JavaScript that makes it easy and rapid to create user interfaces for websites and web apps. It is free and open source. Facebook and a group of independent developers manage this library. With React as a foundation, single-page apps may be developed. Even while React's primary objectives are rendering to the Document object model (DOM) and state management, it is nonetheless more efficient since it employs virtual DOM.

The features of React are as follows:

* Better performance due to the use of Virtual DOM
* Simple dynamic page creation
* Components reusability
* Data flow in a single direction

## **5.4 Visual Studio Code**

An Integrated Development Environment (IDE) for creating and editing source code is called Visual Studio Code (VS Code). In Visual Studio Code, powerful developer tools like IntelliSense code completion and debugging are integrated with the simplicity of a source code editor. An editor prioritizes getting out of your way [8]. The pleasantly frictionless edit-build-debug cycle frees up more time for you to implement your ideas instead of wasting it fiddling with your surroundings. Visual Studio Code supports Windows, Linux, and macOS, so you can get started on any platform right now.

## **5.5 Backend (Node.JS)**

The V8 JavaScript engine from Google Chrome serves as the foundation for Node.js, an open-source, cross-platform runtime environment used to create networking and server-side applications. With the help of the Node.js runtime, JavaScript applications may be developed and executed on Linux, MacOS, and Windows. Moreover, Node.js offers a vast array of different JavaScript modules, which significantly improves the efficiency of building web applications with Node.js. When executing Node.js server processes, the following sequence is observed:

* A server is given a task.
* The server sends the task to the file system.
* The request is awaited by the system.
* The system receives the request, processes the file, and then sends it to the user.

## **5.6 MongoDB**

Operating on a client-server basis, MongoDB is an open-source NoSQL database management system (DBMS) focused on documents. MongoDB stores data in flexible, JSON-like documents as opposed to conventional relational database management systems (RDBMS), enabling more dynamic and schema-less data structures. It is appropriate for managing a variety of data kinds and massive amounts of data as it offers scalability and flexibility for contemporary applications.

***CHAPTER 5***

# **IMPLEMENTATION**

## **5.1 Overview**

A Web project's implementation. To support web applications, backend tools such as databases and APIs are used.

## **5.2 Implementation**

The initial step in the implementation process was downloading and installing the required development tools and packages, such as MySQL, VSC, ES6 (ECMAScript 6), JavaScript, Node.js, Minimal libraries, and React.js.   
As indicated by the ER Diagram, database entities are created following the installation of all tools and packages. Now that MySQL has all the entities, you can begin writing code by creating the login page first.   
The Minimal template was used to develop the project's user interface. The React library is used in combination with HTML, CSS, and JavaScript to generate the system's interfaces.

## **5.3 Front End Using React JS**

React is a component-based framework for JavaScript that makes it easy and rapid to create user interfaces for websites and web apps. It is free and open source. Facebook and a group of independent developers manage this library. With React as a foundation, single-page apps may be developed. Even while React's primary objectives are rendering to the Document object model (DOM) and state management, it is nonetheless more efficient since it employs virtual DOM.

The features of React are as follows:

* Better performance due to the use of Virtual DOM
* Simple dynamic page creation
* Components reusability
* Data flow in a single direction

## **5.4 Visual Studio Code**

An Integrated Development Environment (IDE) for creating and editing source code is called Visual Studio Code (VS Code). In Visual Studio Code, powerful developer tools like IntelliSense code completion and debugging are integrated with the simplicity of a source code editor. An editor prioritizes getting out of your way [8]. The pleasantly frictionless edit-build-debug cycle frees up more time for you to implement your ideas instead of wasting it fiddling with your surroundings. Visual Studio Code supports Windows, Linux, and macOS, so you can get started on any platform right now.

## **5.5 Backend (Node.JS)**

The V8 JavaScript engine from Google Chrome serves as the foundation for Node.js, an open-source, cross-platform runtime environment used to create networking and server-side applications. With the help of the Node.js runtime, JavaScript applications may be developed and executed on Linux, MacOS, and Windows. Moreover, Node.js offers a vast array of different JavaScript modules, which significantly improves the efficiency of building web applications with Node.js. When executing Node.js server processes, the following sequence is observed:

* A server is given a task.
* The server sends the task to the file system.
* The request is awaited by the system.
* The system receives the request, processes the file, and then sends it to the user.

## **5.6 MongoDB**

Operating on a client-server basis, MongoDB is an open-source NoSQL database management system (DBMS) focused on documents. MongoDB stores data in flexible, JSON-like documents as opposed to conventional relational database management systems (RDBMS), enabling more dynamic and schema-less data structures. It is appropriate for managing a variety of data kinds and massive amounts of data as it offers scalability and flexibility for contemporary applications.

***CHAPTER 6***

# **TESTING**

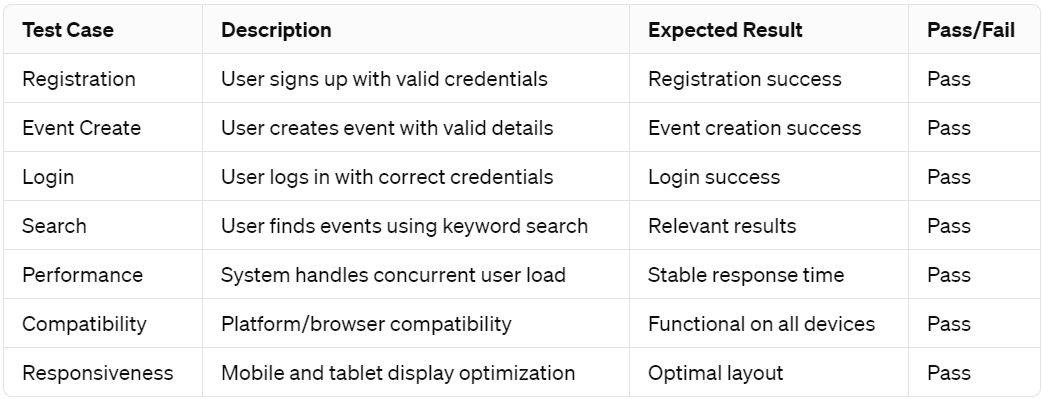
## **6.1 Overview**

In this section, different test approaches are discussed, which were used to test the web after the project's implementation. The backend tools, such as databases were also tested and evaluated.

## **6.2 Test Classifications**

* **Unit Testing**: To assess the many parts and sections of its software architecture, Eventify uses unit testing. By using this testing technique, developers can confirm that each unit behaves correctly and that it performs as expected when used alone. Unit testing allows Eventify to find and address problems early in the development process, resulting in more stable and high-quality code.  
    
  **System Testing:** In order to verify Eventify's general functioning across many platforms and environments, it is put through system testing. This kind of testing evaluates the system holistically, looking at how it behaves and interacts with other elements. System testing helps identify any flaws or anomalies that may occur during execution, ensuring that Eventify satisfies its criteria and operates flawlessly in real-world circumstances.
* **Performance Testing:** To assess Eventify's scalability, responsiveness, and speed under many circumstances, it is put through performance testing. Using this testing technique, the platform's ability to handle user interactions, analyze data, and manage resources is evaluated. Developers may improve Eventify's speed and user experience by using performance testing to find any performance bottlenecks, such as sluggish response times or system failures.  
    
  **Integration Testing:** To ensure that its many modules, components, and external systems work together and are compatible, Eventify goes through integration testing. By identifying any integration problems or inconsistencies, this testing technique makes sure that individual components work as intended when integrated into the bigger system. By guaranteeing smooth data interchange and communication across Eventify's several components, integration testing aids in validating the platform's overall operation and compatibility.

## **6.3 Test Cases**



*Table 6.3: Test Cases*

***Chapter 7***