**

**TRIBHUVAN UNIVERSITY**

**Texas International College**

**Mitrapark, Kathmandu, Nepal**

**A Report**

**On**

**“Slate”**

**Submitted By:**

**Aman Maharjan (15563/074)**

**Kilesh Maharjan (15591/074)**

**Upendra Adhikari (15605/074)**

**A Report Submitted in partial fulfillment of the requirement of Bachelor of Science in Computer Science & Information Technology (BSc.CSIT) 7th Semester of Tribhuvan University, Nepal**

Supervisor’s Recommendations

I hereby recommend that the report prepared under my supervision byAman Maharjan (TU Exam Roll No. 15563/074), Pushpa Pandey (TU Exam Roll No. 15591/074) and Upendra Adhikari (TU Exam Roll No. 15605/074)entitled “**SLATE**” in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology be processed for evaluation.

---------------------------------

**Mr. Rom Kant Pandey**

Project Supervisor

Certificate of Approval

This is to certify that this project prepared by Aman Maharjan (TU Exam Roll No. 15563/074), Pushpa Pandey (TU Exam Roll No. 15593/074) and Upendra Adhikari (TU Exam Roll No. 15605/074) entitled “Slate” in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science and Information Technology has been well studied. In our opinion, it is satisfactory in the scope and quality as a project for the required degree.

-------------------------------------

**Mr. Kumar Poudel**

HOD, Department of CSIT

Texas International College

-------------------------------------

**Mr. Rom Kant Pandey**

Project Supervisor

-------------------------------------

External Examiner

Tribhuvan University

Kirtipur, Nepal

Acknowledgement

We would like to express our deepest appreciation to all those who provided us the possibility to complete this report. A special gratitude we would give to our project supervisor, **Mr. Romakant Pandey**, whose contribution in stimulating suggestions and encouragement, helped us to coordinate our project and writing this report.

Furthermore, we would also like to acknowledge with much appreciation the crucial role of department head **Mr. Kumar Poudel** and staff of B.Sc. CSIT department, who gave permission to use all required equipment and the necessary materials to complete the task. A special thanks goes to constant support from our seniors and every teaching staff of B.Sc.CSIT department which helped us successfully complete our project. Last but not leastwe would like to thank our parents for their love and support, which motivate us to achieve our goal**.** We have to appreciate the guidance given by other supervisors as well as the panels especially in our project presentation that has improved our presentation skills thanks to their comment and advice.

Abstract

Slate is a web based collaboration as well as learning platform for any kind of user to directly interact with the group or an individual by the group formation and video chats also with the integration of Whiteboard with its drawing tools specifically with a core feature of data optimization.

The main objective of this project is to provide an information sharing, communication as well as learning platform and delivering a concept of Whiteboard and also to reduce the data usage by using numerous algorithms based on data optimization such as LZW compression algorithm.

Existing systems such as Zoom, Teams, even Facebook and Viber were studied before beginning the project to find out how learning could be improved in these platforms and this system is built to overcome those findings.

System is built by using one of the prime ‘MERN stack’ besides using individual algorithms. The software has been designed with HTML5, CSS3 & REACT JS with MUI as frontend, NODEJS & MONGODB as backend system and Socket.IO for communications.

Various tools and technologies are used for the development of the application. The software model is based on the agile model . XP methodology As for the tools, we have used software like Git, Zira, Balsamiq, etc.

Testing tools such as JEST and ENZYME are used for unit testing for both NODEJS(Backend) and REACTJS(Frontend). Manual testing was also done by developers.

This is a web based application with a user friendly UI and anyone across the globe can access it.

Table of Contents

[Chapter 1: Introduction 1](#_Toc98966844)

[1.1 Introduction 1](#_Toc98966845)

[1.2 Problem Statement 1](#_Toc98966846)

[1.3 Objectives 2](#_Toc98966847)

[1.4 Scopes and Limitations 2](#_Toc98966848)

[1.4.1 Scope (what to write here) 2](#_Toc98966849)

[1.4.2 Limitations 2](#_Toc98966850)

[1.5 Development Methodology 2](#_Toc98966851)

[1.6 Report Organization 3](#_Toc98966852)

[Chapter 2: Background And Literature Review 4](#_Toc98966853)

[2.1 Background Study 4](#_Toc98966854)

[2.2 Literature Review 4](#_Toc98966855)

[Chapter 3: System Analysis 6](#_Toc98966856)

[3.1 Requirement Analysis 6](#_Toc98966857)

[3.1.1 Functional Requirement 6](#_Toc98966858)

[3.1.2 Non-Functional Requirements 9](#_Toc98966859)

[3.2 Feasibility Analysis 10](#_Toc98966860)

[3.2.1 Technical Feasibility 10](#_Toc98966861)

[3.2.2 Operational Feasibility 10](#_Toc98966862)

[3.2.3 Economic Feasibility 11](#_Toc98966863)

[3.2.4 Scheduling Feasibility 12](#_Toc98966864)

[3.3 Object Oriented Analysis 14](#_Toc98966865)

[3.3.1 Class Diagram 14](#_Toc98966866)

[Chapter 4: System Design 15](#_Toc98966867)

[4.1 System Architecture and Overview 15](#_Toc98966868)

[4.1.1 System Architecture 15](#_Toc98966869)

[4.1.2 System Overview 16](#_Toc98966870)

[4.2 Algorithm Used 17](#_Toc98966871)

[4.2.1 LZW compression algorithm for data optimization 17](#_Toc98966872)

[i. LZW ENCODING PSEUDOCODE 17](#_Toc98966873)

[ii. LZW DECODING PSEUDOCODE 18](#_Toc98966874)

[4.3 Design 19](#_Toc98966875)

[4.3.1 Use Case Diagram 19](#_Toc98966876)

[4.3.2 Activity Diagram 20](#_Toc98966877)

[4.3.3 Class Diagram 21](#_Toc98966878)

[4.3.4 Sequence Diagram 22](#_Toc98966879)

[4.3.5 Component Diagram 23](#_Toc98966880)

[4.3.6 Deployment Diagram 24](#_Toc98966881)

[Chapter 5: Implementation And Testing 25](#_Toc98966882)

[5.1 Implementations 25](#_Toc98966883)

[5.1.1 Tools Used 25](#_Toc98966884)

[Frontend Tools 25](#_Toc98966885)

[Backend Tools 25](#_Toc98966886)

[Database 25](#_Toc98966887)

[Others 26](#_Toc98966888)

[Testing Tools 26](#_Toc98966889)

[5.1.2 Implementation details of modules 26](#_Toc98966890)

[5.2 Testing 27](#_Toc98966891)

[5.2.1 Test cases for Regression testing 27](#_Toc98966892)

[5.2.2 Unit testing and Test cases 31](#_Toc98966893)

[Chapter 6: Conclusion 34](#_Toc98966894)

[6.1 Conclusion 34](#_Toc98966895)

[6.2 Future Recommendations 34](#_Toc98966896)

[Appendix 37](#_Toc98966897)

List of Figures

[Figure 1 Scrum process in Slate. 3](#_Toc98966917)

[Figure 2Use Case Diagram of Login / Register 7](#_Toc98966918)

[Figure 3 Use Case Diagram of Group 8](#_Toc98966919)

[Figure 4 Use Case Diagram of Video call and Whiteboard 9](#_Toc98966920)

[Figure 5 Gantt Chart 13](#_Toc98966921)

[Figure 6 High Level Class Diagram for Slate 14](#_Toc98966922)

[Figure 7 Higher level System Architecture 15](#_Toc98966923)

[Figure 8 Lower-level System Architecture 15](#_Toc98966924)

[Figure 9 Use case diagram for slate 19](#_Toc98966925)

[Figure 10 Activity Diagram for slate 20](#_Toc98966926)

[Figure 11 Class Diagram for Login/Registration in slate 21](#_Toc98966927)

[Figure 12 Class Diagram for Group module in Slate 21](#_Toc98966928)

[Figure 13 Sequence diagram for Slate 22](#_Toc98966929)

[Figure 14 Component Diagram for Slate 23](#_Toc98966930)

[Figure 15 Deployment diagram for Slate 24](#_Toc98966931)

[Figure 16 Backend unit testing results 31](#_Toc98966932)

[Figure 17 Frontend unit testing results 32](#_Toc98966933)

List of Tables

[Table 1 Functional Requirement 6](#_Toc98966978)

[Table 2 Yearly expexted income of Slate 11](#_Toc98966979)

[Table 3 Time Schedule 12](#_Toc98966980)

[Table 4 Test Case for Registration 27](#_Toc98966981)

[Table 5 Test cases for Login 29](#_Toc98966982)

[Table 6 Test Case for Group 30](#_Toc98966983)

[Table 7 Test cases for Notification 31](#_Toc98966984)

List of Abbreviations

**CASE -** Computer-aided software engineering

**CSS -** Cascading Style Sheet

**HTML -** Hypertext Markup Language

**ICT -** Information’s And Communications Technology

**LZW -** Lempel-Ziv-Welch

**MERN -** MongoDB Express React NodeJs

**PIECES -** Performance Information Efficiency Control Economics Services

**SASS -** Syntactically Awesome Style Sheets

**TV -** Television

**UI -** User Interface

**UNESCO -** United Nations Educational, Scientific and Cultural Organization.

# Introduction

## Introduction

Evolution in technology is essential to unravel the problems and challenges in the world. Also, we are constantly evolving as we are interacting with the world. So, in the account of this, it all began when there was the most perilous pandemic of Novel Coronavirus, also known as COVID-19 which has disrupted many aspects of human life, including education. According to a UNESCO report, 1.6 billion children across 191 countries have been largely impacted by the temporary closure of educational institutions . To mitigate this impact, many educational institutes have responded with their own way of solution. Most solutions included ICT technology in order to provide continuity in educational activity. As online education relies on technology which includes the internet and Wi-Fi, the discrepancies that exist in their availability are widening the gaps in access and quality of education. Government also tries to use traditional technologies such as radio and TV but when it comes to learning one way communication is not enough. The pandemic has had serious impacts on students’ learning and well-being, and that it potentially widens the gaps between advantaged and disadvantaged children in their equitable access to quality education.

On account of that we have lined up a scheme designed to compose a strong, light and effective platform for participation and communication between two or more remotely placed individuals or groups named Slate. The main feature of our product Slate is to provide a learning platform for any kind of user to directly interact with the group or an individual by use of Whiteboard along with its drawing tools specifically with a core feature of data optimization not to be overlooked. We tend to use one of the prime ‘MERN stack’ besides using individual algorithms. We are providing students/or any kind of learning alacrity loaded groups of people with our special feature integrated Whiteboard where the individuals will get the latest facilities at the least uses of data packets where there won't be any issue of packet loss while accessing our Web App .So, overall Slate is a persistent chat-based collaboration platform complete with online meetings, group formation including video chat and many more extremely useful features for communications and learning. There certainly are quite a lot of options to choose from. But one particularly useful solution stands out and is not obsolete is: Slate.

## Problem Statement

Concerning the current scenarios, our whole country is shifting to online education. In parallel, the current pandemic is impacting rural students more than those who live in cities. While many are making the most of what online learning offers, local authorities are finding it tough to integrate high-speed internet facilities. Furthermore, purchasing data plans for learning could also incur a lot of expenses for families who face financial constraints which have affected the participation levels of both teachers and students concerning live classes. So, to resolve this kind of problem we are introducing our web app i.e. Slate, a free learning & interacting platform to settle the issue of data management while conducting online interaction.

## Objectives

The objectives of the system are:

* To provide an information sharing, communication as well as learning platform and delivering a concept of Whiteboard.
* To reduce the data size during the exchange of learning material between the sender and receiver.
* ……………

## Scopes and Limitations

### Scope (what to write here)

The scopes of the system are:

* To
* To.
* To.

### Limitations

The limitations of the system are:

* Limited features are available in the group video call like screen share facility is not available.
* Is only accessible via browser as we are working on a web app for now.
* Storage is limited as we are using free resources for now.

## Development Methodology

The scrum of Agile methodology is used for project management that divides a project into backlogs. It's a project management method that entails ongoing collaboration and iteration, allowing us to deliver value to clients more quickly and with fewer headaches. Work is produced in small, digestible amounts with scrum. Requirement plans and results are evaluated on a regular basis through stand-ups. Changes are progressive and evolutionary rather than revolutionary with user engagement, thus it can be useful in promoting change that is important to the success of most project transitions. As a result, the scrum development methodology is a suitable decision.

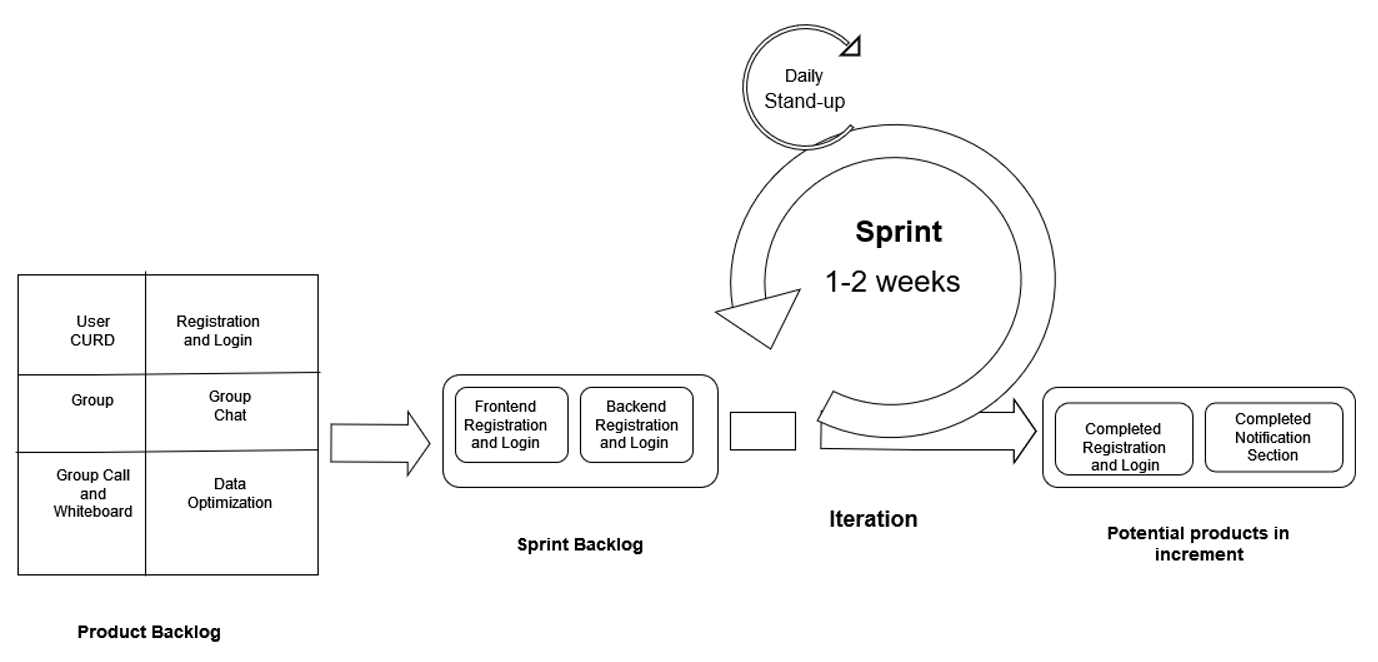


Figure Scrum process in Slate.

## Report Organization

The report is based on six chapters. The project's resources, requirements, and diagrammatic representations are divided into six chapters. Each chapter follows the creation of our project from start to finish.

The first chapter gives a general summary of our project. It foreshadows and includes key concepts that will be covered in the following chapters.

The project's background study and literature evaluation are detailed in Chapter 2, which follows the introduction chapter. It enables you to distinguish between different hypotheses, strategies, and problems in the current investigation.

The third chapter covers a lot of ground when it comes to system analysis. It includes a requirement analysis, a feasibility analysis, and object-oriented analysis. It outlines each experiment's goals and addresses the difficulties presented in Chapter 3.

For the system design in Chapter 4, an object-oriented technique is employed, which involves the refinement of class, state, object, sequence, and activity diagrams. Diagrams of components and deployment are also supplied. This helps you understand the system design and algorithm that was used to create it.

In Chapter 5, a number of CASE tools, as well as computer languages and database platforms, are briefly discussed. In this sub-topic, the implementation details of modules are defined. During the testing phase, unit and system testing are performed to assure accuracy. This chapter discusses the results of the tests.

In Chapter 6, the project's findings and future recommendations are offered. This gives an indication of how the project will be evaluated as a success and what improvements might be made.

# Background And Literature Review

## Background Study

The global pandemic has created a shift in how people work and learn a shift that will become the new normal as we all figure out how to function in a hybrid world. The last 2 years have seen the evolution of hybrid work develop into a reality being adopted across the globe. So we attempted our best to coordinate with Whiteboard with the online instruction with least amount of data packet loss & low bandwidth requirement additionally free of cost. Talking about the study perspective of this project, the Slate stroked into our mind just by observing the current havoc happening in the world. The situation of Covid is becoming a never ending part which is severely affecting the educational sector as well as student’s mentality. data

So, the problem was discussed by observing the situation of children who are financially in reverse as well as distant from the reach of surplus data. Education is one of the foremost imperative portions of our life after our prosperity and wellbeing. Even Though the people living inside the city areas may not find it difficult to attend the online classes in these difficult situations, by observing the situation of the children of rural areas through various mediums such as televisions, radios and even by our own research we discovered the circumstances exceptionally vital. Some children are finding a different way. For instance, we might have heard the news of a child who built a house on the tree for his online class due to network problems as well as the platform where he or any other children get that from might consume a lot more data bandwidth. For that he is risking his life. As a matter of concern we have decided to build such a light weighted learning platform which consumes less data as compared to existing software’s such as MS Teams, Zoom and many more also with integrated Whiteboard interior in it.

## Literature Review

As mentioned above, there are numerous platforms that benefit peoples in the real world by the aid of virtual meetings also considering the education sector. As far as our concern, there are few platforms which tumble under our research section considering the similar concepts with our project. One of them is Zoom. Zoom was originally founded in 2011. The pandemic has kick started this system in a wide range. At the same time Zoom has been criticized for "security lapses and poor design choices" that have resulted in heightened scrutiny of its software.

Getting into another similar kind of software, Microsoft Teams is also widely time-honored software.It was created during an internal hackathon at the company headquarters. Microsoft Teams is simply the collaboration app that brings together all the diverse functionalities and tools in one place for simple communication, fast collaboration and widespread integration with external apps and bots.

By studying existing systems such as Zoom[[1]](#footnote-1) ,Teams, even Facebook and Viber we find out that most of the system consists of some of the use cases that are implemented in the Slate. Some got some security issues while others got more complex to normal users. There are so many of them yet users are still very much confused as to which one they have to use in different circumstances. So it is not the axiom that if there exists some sort of system that already existed then, there also can be a similar kind of improvised system which may appeal better also. We can take good examples from Facebook. Although there already was Facebook universally accepted but eventually new software’s such as Instagram, Twitter, Whatsapp get positioned in the market successfully. Likewise our system has features tending to solve the big rising issue namely digital divide as it is mostly concerned with the bodies that are behind the reach of excessive data commonly in remote areas .To overcome this problem our web application will have an interactive whiteboard where both teacher and student can interact by optimizing the data needed for online communication along with features of traditional learning platforms like sharing video and other resources.

# System Analysis

## Requirement Analysis

E-learning has evolved over the last few years from being a supplementary tool to becoming an essential companion to teaching, however many platforms that provide these services do not provide tools that student and teacher want for betterment of the class or learning. Taking that in mind we conducted an online survey for the requirement analysis of this platform and found out different features are provided in different platforms but there is not any good platform to provide all of the services and tools which are necessary for the effective class. This gathering requirement aids in determining our users demands, determining project size, and determining the timeframes and resources required to finish the project.

(data source for need for this project)

### Functional Requirement

Functional requirements of Slate deal with what the system provides for users. Different functional requirement are given below:

Table Functional Requirement

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SN | Name | Use Case | Action/Process | Constraints | Dependency |
| 1. | Sign Up | Users need to create an account to use the system. | Fill out the Sign-Up form  Username:  Email:  Password: | Should not leave the required field empty | Must have valid email id for email verification. |
| 2. | Login | User that has already created an account in our platform can log in. | Provide Email and password | Must provide authentic credentials | Must have confirmed email |
| 3. | Group | Users should be able to create and update Groups. | Provide group name and description | Should not leave the required field empty | Login |
| 4. | Join Group | Users can invite other user to join group and invited user can join through link. | Select users to invite. | User should have registered. | Sign Up ,Group , Login |
| 5. | Group Video Call and Chat | Users should be able to chat and make video calls. | Type a message and send it to chat . click on the call button to make a call. | Users should be in the group. | Login , Group,Join Group. |
| 6. | Whiteboard Sharing | Users should be able to share whiteboards and draw. | Should click share whiteboard and shart to draw. | Users should have permission to use the whiteboard. | Group video call. |

The following use case diagram represents the working of our system. In the given diagrams, functionality of above mentioned requirements of the system (Slate) are described.

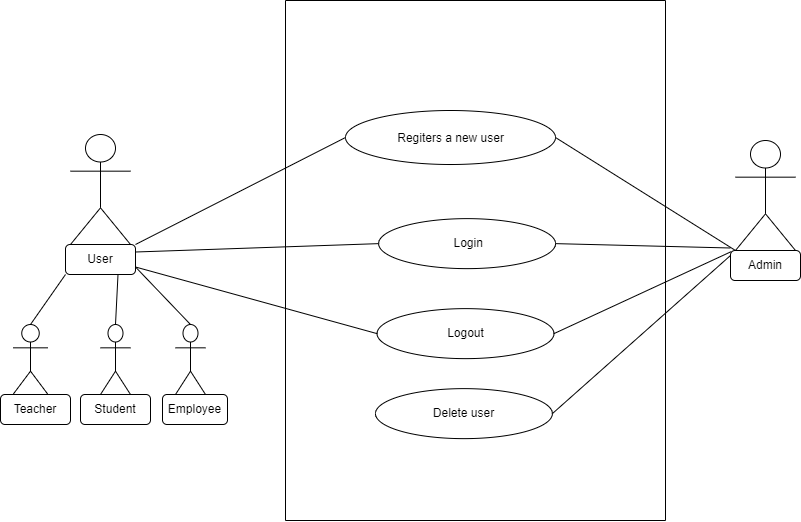
****

Figure Use Case Diagram of Login / Register

This Use Case diagram above illustrates the User registering for the first time within the application It also illustrates the logout, if the details are correct. And the processes are visually represented where the activities are assigned according to the role of an actor where the deletion of the user is authorized by the admin itself.

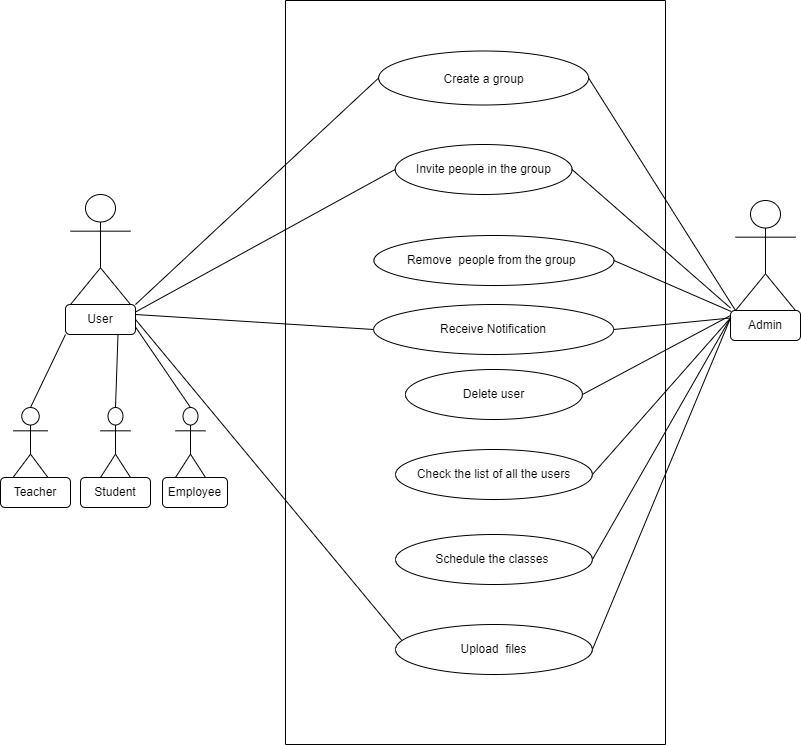


Figure 3 Use Case Diagram of Group

This Use Case diagram above illustrates the user creating the group and inviting other users via email .Here the invitation is notified through the notification on the application and also users can upload files in their respective group. The role of removing the users is given to the admin.



Figure ***Use Case Diagram of Video call and Whiteboard***

This Use Case diagram above illustrates the process where the user starts the group call and can chat with group members where the actors can share their video. The admin will have access to the Whiteboard along with its drawing tools. The user can access the calendar whereas the admin can also access the statistical report of the compressed data.

### Non-Functional Requirements

1. **System Reliability Requirements:**

* The system is dependable.
* If the request cannot be processed, an appropriate error message is displayed.
* Web pages load in a matter of seconds.

1. **Requirements for safety**

* It is necessary to keep track of its specifics.
* Authentication of users is required.
* Database is backed up on a regular basis.

1. **Requirement for security**

* The user can view his profile after providing his password and user id.
* The information of the users is kept safe and secure.
* Details are not shared.

## Feasibility Analysis

### Technical Feasibility

As this system is completely web based. So hereby implementation and working of this system is technically feasible on any devices with any modern browsers. This application requires a minimum of Windows XP platform with a high configuration of 1GB RAM and an Intel Pentium Dual Core CPU. This is technically doable.

Each of the above mentioned technologies are available and our team has the technical skills required to develop the project. Implementation of this project is within the time limitations of the development. For now, the website is hosted in a free web hosting space, but for later implementations it will be hosted in a paid web hosting space with sufficient bandwidth. From these it’s clear that the project is technically feasible.

### Operational Feasibility

The proposed system follows PIECES framework which helps in identifying problems to be solved and their urgency:

**Performance:**

With the implementation of image compression database normalization, the system will provide adequate throughput and response time.

**Information:**

The system includes optimized form to get maximum information which will provide end users and managers with timely, pertinent, accurate, usefully formatted information.

**Economy:**

Initially this system will be hosted in a free hosting system to provide cost effective information to the service.

**Control:**

To offer effective controls to protect against fraud and guarantee security of data and information the system collects user details for user identification and verification.

**Efficiency:**

This system maximizes the use of available data and resources Services: The proposed system will be initially in testing phase with a limited area and later will grow globally. So, the system will provide reliable service and will be flexible and expandable for future work.

### Economic Feasibility

Technologies that are being used to develop the project are freely available and initially the web site is hosted in a free web hosting space. So, the system is economically feasible.

Here, analysis is done on the basis of cost/benefit. For this analysis, determine the benefits and savings that are expected from a candidate system and compare them with costs. For this analysis we use ROI analysis using given formulae.

i)Expected cost for first six month (Development time):

System Analysis and Requirement Analysis and Design cost = 30,000

 UI and frontend designs cost = 50,000

 Development cost per month = 25,000

ii)Expected cost for Hosting and Database (1 year cost) = Rs 2,600 + Rs 2,000 = Rs 24,000

iii) Expected cost for next six month (Marketing and Enhancement Time):

Marketing cost per month = 7,000

Maintenance and enhancement cost per month for first year = Rs 20,000

Total expected Investment in first year = 30,000 +50,000 + (25,000\*6) + 24,000 + (7000\*6) + (20,000\*6) = Rs 3,08,000

Expected cost for service (average 13 school to use service in initial year with average 500 student) = 32,000

Expected maintenance and enhancement cost per month from second year = 20,000

Total expected Income per year with development charge = (32,000\*13) - (12\*20000) = 1,76,000

Yearly Income with 20% increase in the service:-

Table Yearly expexted income of Slate

|  |  |
| --- | --- |
| Year | Income |
| 0 | - 3,08,000 |
| 1 | 1,76,000 |
| 2 | 2,11,200 |

For calculating ROI,

Net profit = (1,76,000 + 2,11,200 - 3,08,000)

 = Rs 79,200

Average Annual profit = 79,200/2

 = Rs 39,600

ROI = 39,600/3,08,000 \* 100%

 = Rs 12.86 %

Payback period = 2years

Hence, we can conclude that our proposed system is economically feasible.

### Scheduling Feasibility

Table Time Schedule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SN** | **Task** | **Time (in days)** | **Start Date** | **End Date** |
| 1 | Background Study | 5 | June 10, 2021 | June 15, 2021 |
| 2 | Requirement Analysis | 5 | June 16, 2021 | June 20, 2021 |
| 3 | System Analysis | 5 | June 21, 2021 | June 25, 2021 |
| 4 | High Level Design | 7 | June 26, 2021 | July 2, 2021 |
| 5 | Detailed Design | 8 | July 3, 2021 | July 11, 2021 |
| 6 | Login/Signup | 9 | July 12, 2021 | July 20, 2021 |
| 7 | Group | 95 | July 21, 2021 | October 25, 2021 |
| 8 | Notification | 14 | October 26, 2021 | November 10, 2021 |
| 9 | Calendar | 9 | November 11, 2021 | November 20, 2021 |
| 10 | Optimization | 15 | November 21, 2021 | December 5, 2021 |
| 11 | Final Testing & Maintenance | 14 | July 12, 2021 | December 20, 2021 |
| 12 | Deployment & Documentation | 15 | July 21, 2021 | January 10, 2022 |

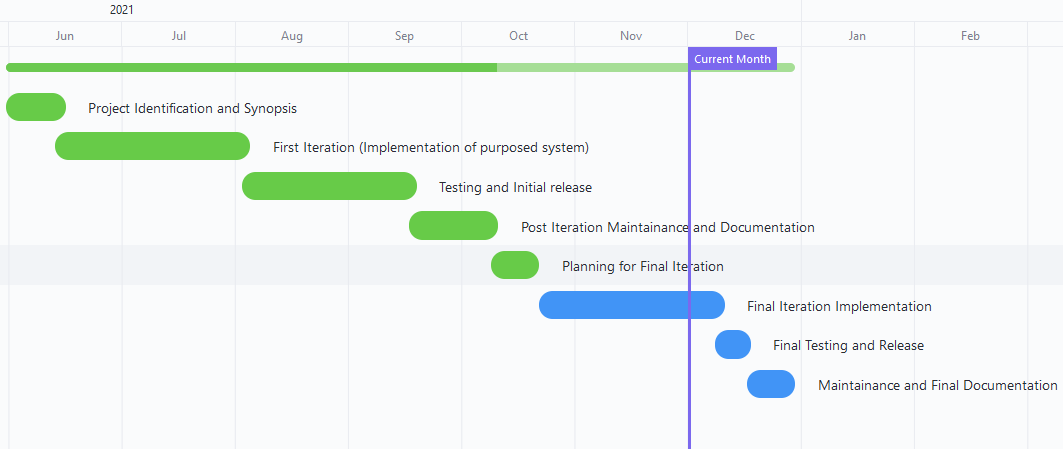
****

Figure Gantt Chart

The Gantt chart for the development plan is given above. The plan explains the tasks versus the time that will take to complete.

The project was officially started on June 10, 2021 with the problem identification and synopsis. The system design was started on January 10, 2022, the coding and implementation activities were carried out parallel with system design from August, 2021 for generic modules like user authentication & authorization. The testing was parallel as TDD approach was preferred. However different testing methods were used and were conducted for a longer period till November, 2021. The documentation phase was carried for the entire period. The project was concluded in January, 2022.

## Object Oriented Analysis

### Class Diagram

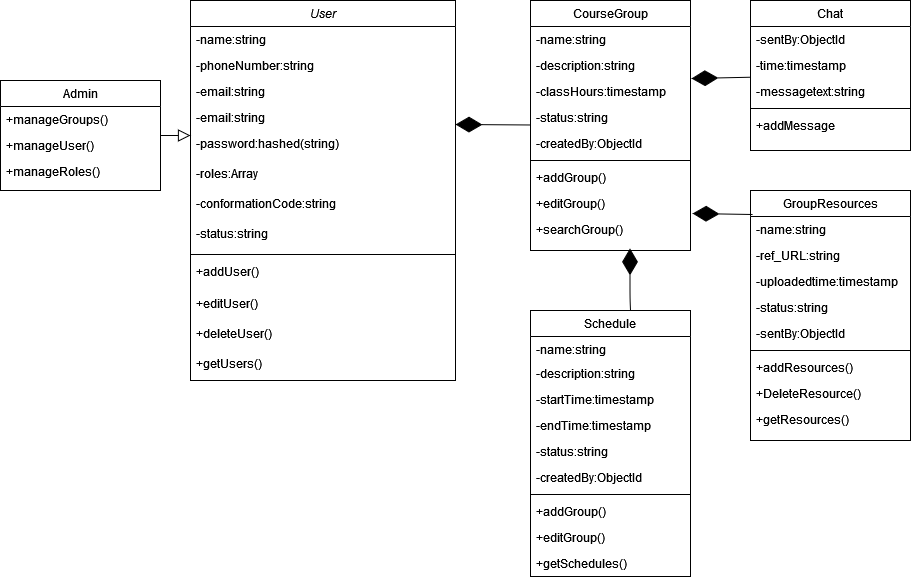
****

Figure High Level Class Diagram for Slate

The above class diagrams show the relationship between each model. Each model extends the Base Model which consists of basic properties like id, createdAt, updatedAt, deletedAt. The user model is categorized into three types, Admin, Teacher and Student which are indicated by the userType property. Similarly, the admin and teacher users are attached with roles consisting of various permissions. The permissions are stored in json type and are used as an authorization.

# System Design

## System Architecture and Overview

### System Architecture

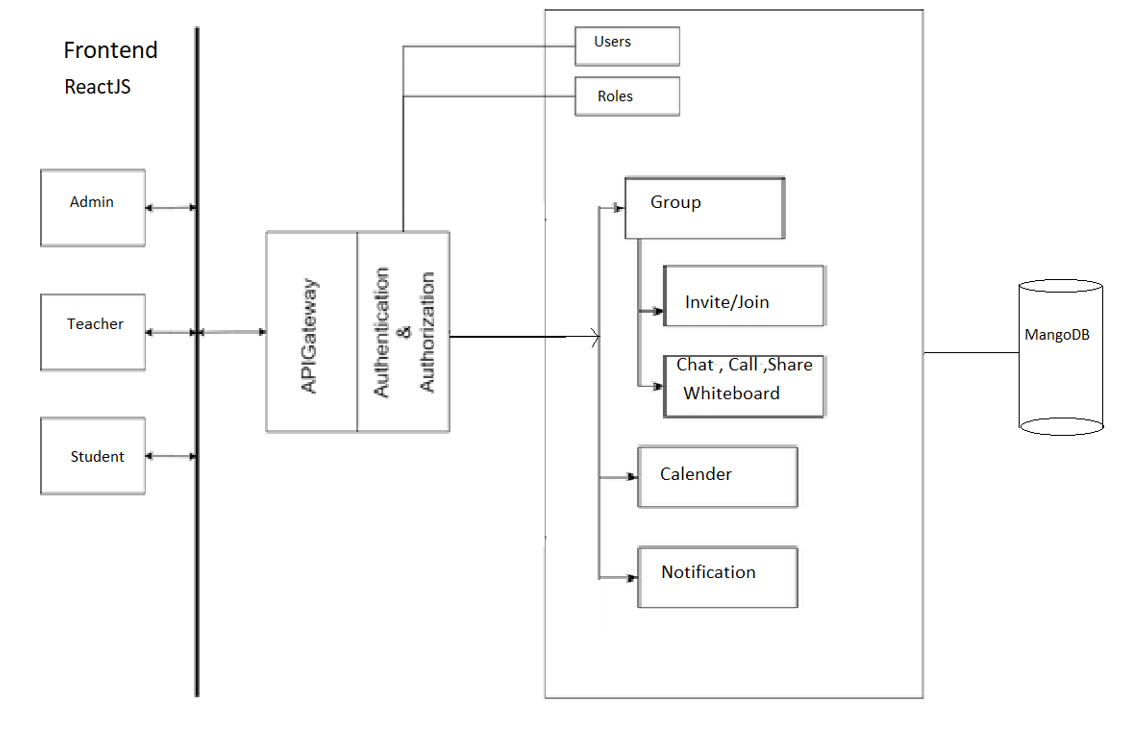


Figure 7 Higher level System Architecture

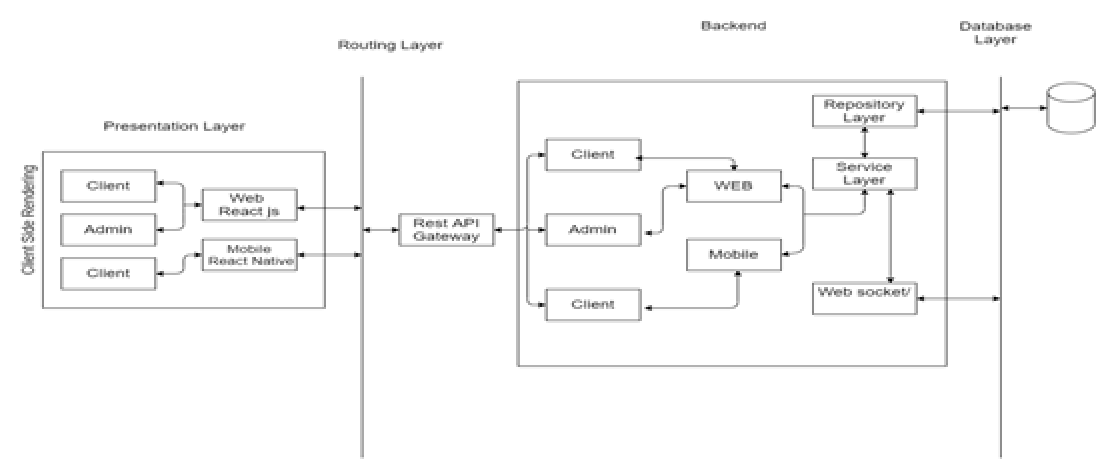


Figure Lower-level System Architecture

### System Overview

The system consists of 4 layers namely:

* **Presentation layer (Frontend)**

This layer consists of client, admin and a vendor interface designed using HTML, CSS, JS and ReactJs. The presentation layer contains user interfaces which can be viewed and navigated by users. The actions performed by the users are passed to the Routing layer.

* **Routing layer (API Gateway)**

This layer handles the incoming requests from the presentation layer and proxies it to the backend. It also consists of user authorization and authentication. The routing layer again forwards the response received from the backend layer to the presentation layer.

* **Backend layer**

It receives all the requests from the presentation layer via the routing layer. In this layer different micro-services are present. All the services are connected to a single database from which data is fetched and updated. The service layer handles the business logic of the application and passes the response to the routing layer.

* **Database layer**

NoSql database system MangoDB is used for storing the system's data. The features provided by this database are suitable for the current system like generation of UUID, storing JSON data and querying them.

## Design

### Use Case Diagram

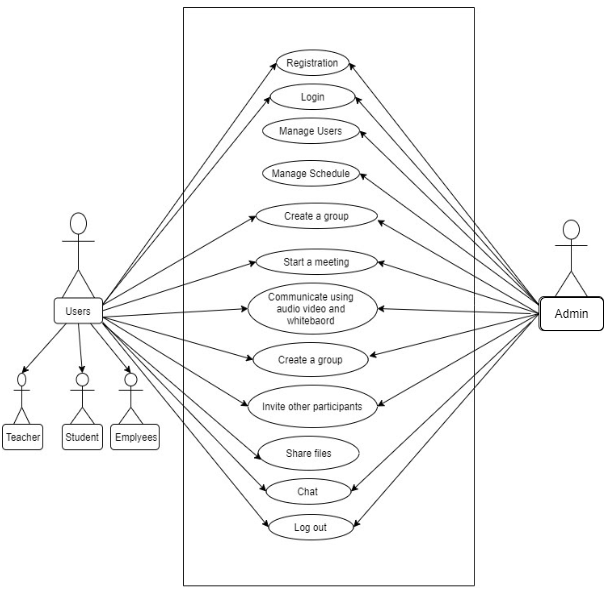
****

Figure Use case diagram for slate

This use case diagram is a visual representation of the process that is required in the application. An user initially creates a group and other users (students) can join in the group to chat, video call and share resources in the group.

### Activity Diagram

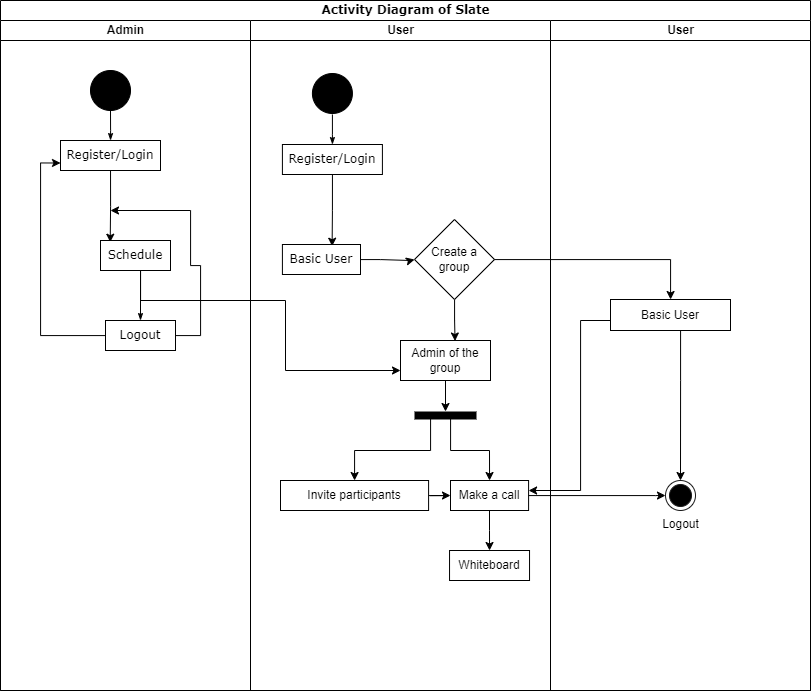
****

Figure Activity Diagram for slate

This diagram shows the activities users can perform in the system. If the user is valid then the user can perform various activities such as joining a group, video calling, sharing whiteboard etc.

### Class Diagram

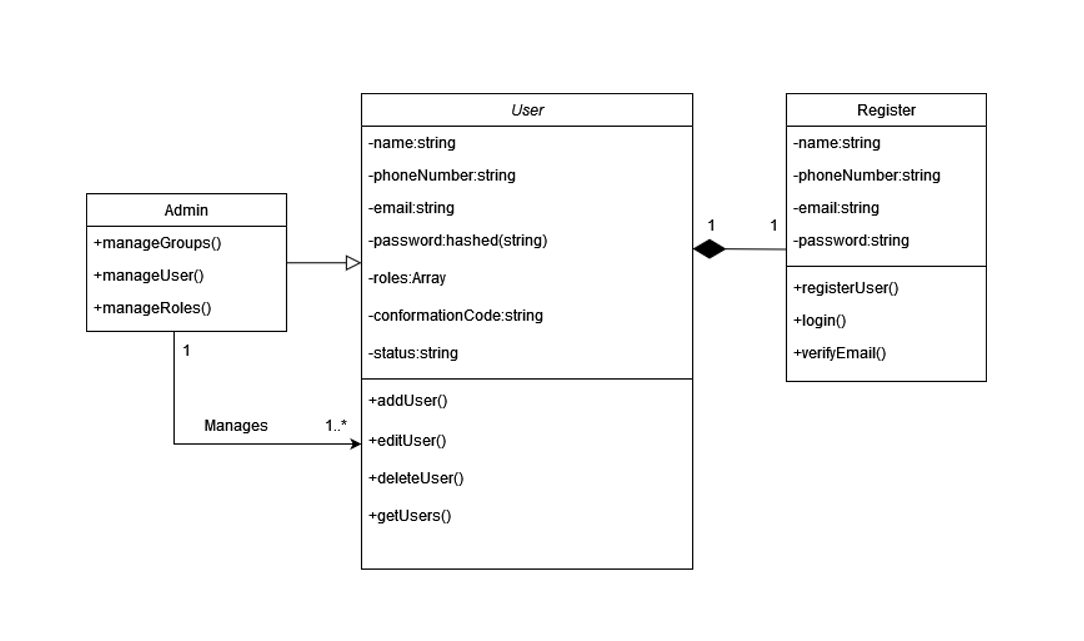
****

Figure Class Diagram for Login/Registration in slate

The above class diagrams show the relationship between users and the registration process.

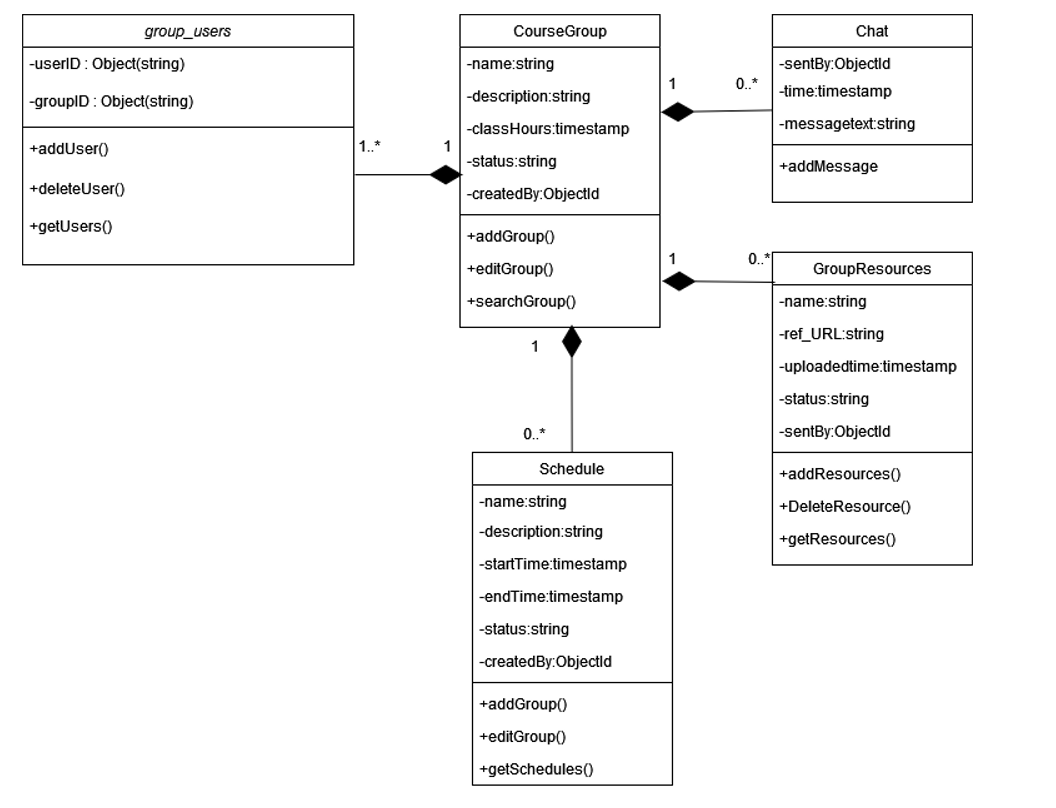


Figure Class Diagram for Group module in Slate

### Sequence Diagram

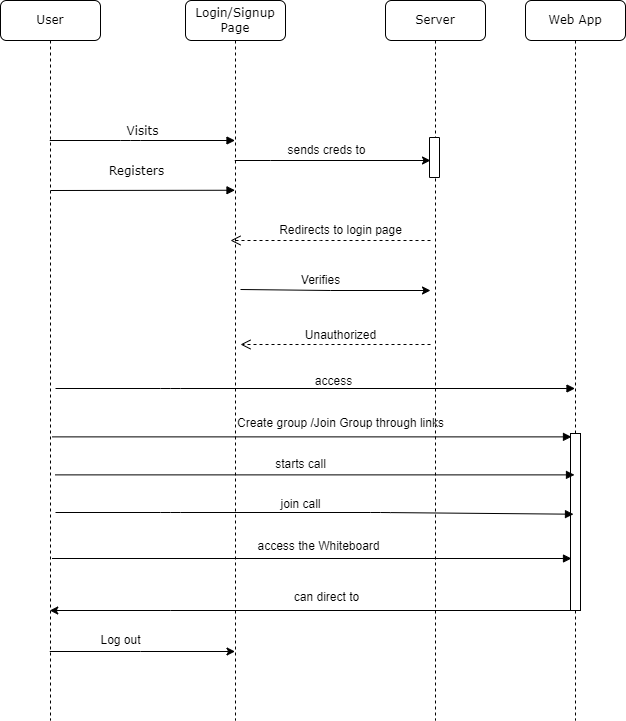


Figure Sequence diagram for Slate

Firstly, the registration process is performed where data is validated and appropriate response is sent back. The user is then redirected to the login page, from which they can login to the system and use other features of the system.

### Component Diagram

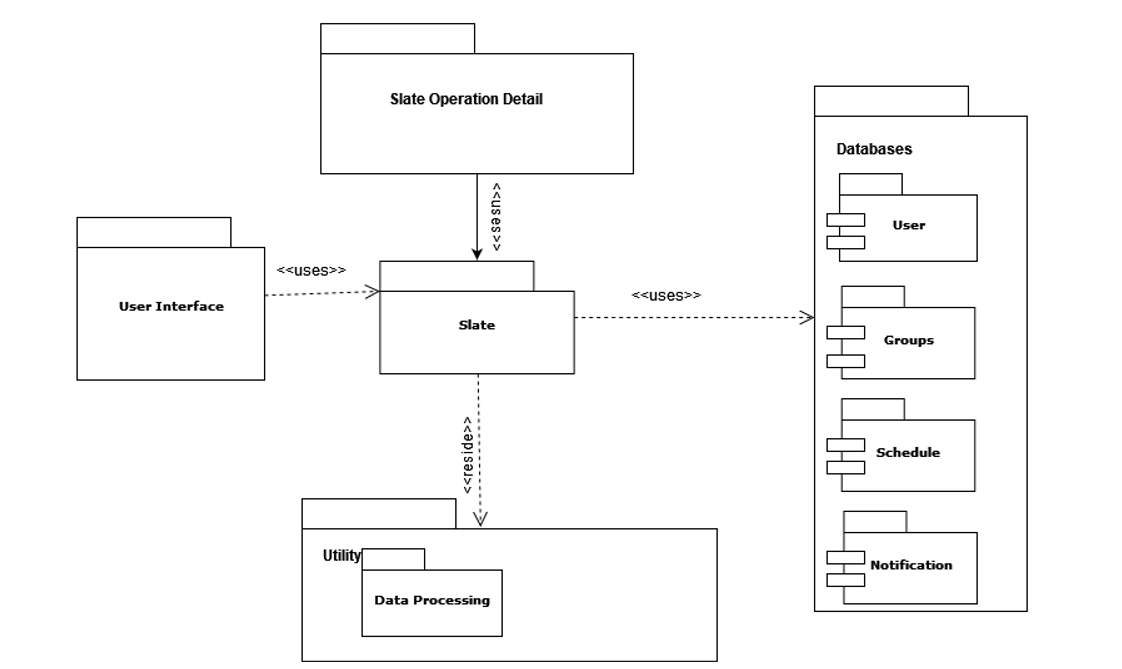


Figure Component Diagram for Slate

Above diagram shows the system's components that model the static implementation view of a system of Slate. It describes a module of classes in slate that represent independent systems or subsystems with the ability to interface with the rest of the system.

### Deployment Diagram

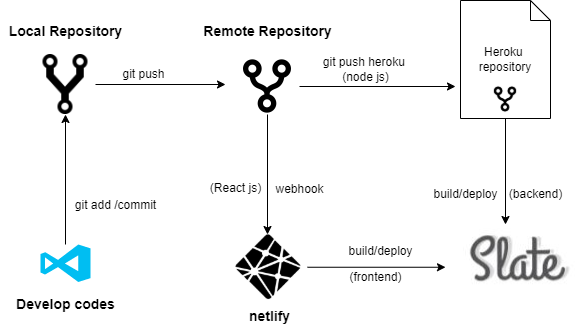


Figure Deployment diagram for Slate

Problems which people face in the IT Industry are long release cycles, tedious management of artifacts, rollbacks, patching software, versioning, etc. Below innovative solution describes methodologies which resolve these issues to a great extent.

**Continuous Integration (CI)** — It is a development practice that requires developers to integrate code into a shared repository several times a day. Each check-in is then verified by an automated build, allowing teams to detect problems early.

**Continuous Deployment (CD)** — It is a strategy for software releases wherein any code commit that passes the automated testing phase is automatically released into the production environment, making changes that are visible to the software’ users.

NodeJs was used to design and develop the micro-service and ReactJs is used as a prime builder for frontend components which displays data provided by services created on NodeJs, when requested. As soon as, developer commits and pushes his/her code in the Git repository, a Pull Request is created where we assign our fellow developer to review the code. This process is essential as it helps to ensure that quality reviewed code is merged in the main repository resulting in bugfree deployment. After the codes are reviewed, it is then merged with the main branch.

After merging with the main branch, node js codes are automatically pushed and merged with the heroku remote repository which then starts the building process where new packages are installed. After the building process is finished, it is then deployed. Similarly, React Js codes are updated in netlify with the help of webhooks which builds and installs new packages. After the building process is finished, it is then deployed as well.

## Algorithm

### LZW compression algorithm for data optimization

The LZW compression algorithm works by reading a sequence of symbols, grouping the symbols into strings, and converting the strings into codes. Because the codes take up less space than the strings they replace, we get compression. Characteristic features of LZW includes,

* LZW compression uses a code table, with 4096 as a common choice for the number of table entries. Codes 0-255 in the code table are always assigned to represent single bytes from the input file.
* When encoding begins the code table contains only the first 256 entries, with the remainder of the table being blanks. Compression is achieved by using codes 256 through 4095 to represent sequences of bytes.
* As the encoding continues, LZW identifies repeated sequences in the data and adds them to the code table.
* Decoding is achieved by taking each code from the compressed file and translating it through the code table to find what character or characters it represents.

#### i. LZW ENCODING PSEUDOCODE

1 Initialize table with single character strings

2 P = first input character

3 WHILE not end of input stream

4 C = next input character

5 IF P + C is in the string table

6 P = P + C

7 ELSE

8 output the code for P

9 add P + C to the string table

10 P = C

11 END WHILE

12 output code for P

#### ii. LZW DECODING PSEUDOCODE

1 Initialize table with single character strings

2 OLD = first input code

3 output translation of OLD

4 WHILE not end of input stream

5 NEW = next input code

6 IF NEW is not in the string table

7 S = translation of OLD

8 S = S + C

9 ELSE

10 S = translation of NEW

11 output S

12 C = first character of S

13 OLD + C to the string table

14 OLD = NEW

15 END WHILE

# Implementation And Testing

## Implementations

### Tools Used

#### Frontend Tools

**HTML5:**

HTML or Hyper-Text Markup Language can be referred to as the Worldwide Web’s primary language. Most of the web pages hosted on the internet are written in some variation of HTML. HTML has seen many updates over time, and currently, the newest HTML version is HTML5. HTML5 provides full support for JavaScript to run in the background; this is possible courtesy to the JS web worker API of HTML5. Hence, the project is developed through HTML5.

**CSS3:**

CSS3 is the advanced level of CSS2.0. In CSS3, new properties - border radius, box shadow, text shadow, multiple background images and much more which are required to create user friendly UI for this web based project. Furthermore, CSS3 is supported by all new browsers.

**JS:**

JavaScript is most commonly used as a client-side scripting language. JavaScript code is written into an HTML page. When a user requests an HTML page with JavaScript in it, the script is sent to the browser and the browser responds with the appropriate actions.

**ReactJS:**

React is a JavaScript library for building user interfaces. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications.Project used this library for the to create a web based single page application through reusable components.

#### Backend Tools

**NodeJS:**

As an asynchronous event-driven JavaScript runtime, Node.js is designed to build scalable network applications , hence it is chosen as a backend tool for the project.

This is in contrast to today's more common concurrency model, in which OS threads are employed. Thread-based networking is used in the project which is relatively inefficient but difficult to use. Almost no function in Node.js directly performs I/O, so the process never blocks except when the I/O is performed using synchronous methods of Node.js standard library. Because nothing blocks, scalable systems are very reasonable to develop in Node.js.

#### Database

**MongoDB:**

MongoDB is a powerful, open-source NoSql database system with many features that safely store and scale the most complicated data workloads which is perfect for our application. Well architectured mongoose, reliability, data integrity, robust feature set, extensibility, and the dedication of the open-source community behind the software to consistently deliver performance and innovative solutions , this will be helpful for the the future updates on the project.

#### Others

**Heroku:(aman)**

Heroku is a cloud platform as a service supporting several programming languages. One of the first cloud platforms, Heroku has been in development since June 2007, when it supported only the Ruby programming language, but now supports Java, Node.js, Scala, Clojure, Python, PHP, and Go.

**Jira Project Management:**

Jira Software is an agile project management tool that supports any agile methodology, be it scrum, kanban, or your own unique flavor. For scrum boards, backlogs, roadmaps, reports, to integrations and add-ons this software is used in management of this project.

**Github:**

GitHub is a code hosting platform for version control and collaboration. Project is developed through collaboration between the developer, different versions of projects are maintained in github and code is merged through git command on regular basis after resolving conflicted code.

#### Testing Tools

**Jest:(aman)**

Jest is a JavaScript testing framework maintained by Facebook, Inc., designed and built by Christoph Nakazawa with a focus on simplicity and support for large web applications. It works with projects using Babel, TypeScript, Node.js, React, Angular, Vue.js and Svelte.

### Implementation details of modules

The proposed system is designed following a micro service architecture consisting of various services like group’s actions, notifications, user authentication and authorization programmed using Node as a backend. Modern application consists of many users which directly results in many requests and responses. Handling such requests in production is very difficult as it requires more hardware resources. Replicating the same application for handling a certain traffic can be done, but replication of the whole application will require more hardware resources. But in an application only a certain feature will be overused i.e. many users will be using a certain part of the application more often. Hence while designing a micro service a clear idea of the proposed system must be done. It is very difficult to choose the granularity of the services. In the application each module is separated so that it can be easily changed into various services during production.

Query is used for fetching data from the backend, it uses Http GET requests.

For creating, updating and deleting an object, it uses Http Post requests.

Different module implemented in the project are described below:

**Group:**

The user can be added to different groups and invitation to join the group is done through email.

**Group Chat:**

Users can chat with users in groups.

**Group Video Call:**

The user video calls on the group to interact and learn with other group members.

**Whiteboard with drawing tool:**

Users can use an inbuilt whiteboard to share their thought drawing.

**Notification:**

(code and picture)

other algo and libraries used in project (where /why)

## Testing

In this testing phase our system as a whole was tested. Every individual component was integrated and tested against user and hardware compatibility. Sometime in this testing process dependency error was found due to different local server environments and dependency conflict. To overcome this problem different third party packages were updated from time to time.

### Test cases for Regression testing

Table Test Case for Registration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SN** | **Test Case ID** | **Test Description** | **Test Data** | **Expected Result** |
| 1 | TC-REG-01 | Registering a user in the application | Fullname:  John Doe  Username:  john\_doe  Email Address:  [johndoe@gmail.com](mailto:johndoe@gmail.com)  Password:  Qwerty@123  Confirm Password:  Qwerty@123 | Registered Successfully |
| 2 | TC-REG-02 | Registering a user with empty input fields | Fullname:  Username:  Email Address:  Password:  Confirm Password: | Fields are required error message should be displayed |
| 3 | TC-REG-03 | Registering a user with invalid Email | Fullname:  John Doe  Username:  john\_doe  Email Address:  [johndoe.com](mailto:johndoe@gmail.com)  Password:  Qwerty@123  Confirm Password:  Qwerty@123 | Invalid email error message should be displayed |
| 4 | TC-REG-04 | Registering a user with weak password | Fullname:  John Doe  Username:  john\_doe  Email Address:  [johndoe@gmail.com](mailto:johndoe@gmail.com)  Password:  Qwerty  Confirm Password:  Qwerty | Weak password error message should be displayed |
| 5 | TC-REG-05 | Registering a user with a different Password and Confirm Password. | Fullname:  John Doe  Username:  john\_doe  Email Address:  [johndoe@gmail.com](mailto:johndoe@gmail.com)  Password:  Qwerty@123  Confirm Password:  Qwerty@1233 | Password doesn’t match error should be displayed |
| 6 | TC-REG-06 | Registering a user with a username that is already registered to the application | Fullname:  John Doe  Username:  john\_doe  Email Address:  [johndoe@gmail.com](mailto:johndoe@gmail.com)  Password:  Qwerty@123  Confirm Password:  Qwerty@123 | User exists error should be displayed. |

Table Test cases for Login

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SN** | **Test Case ID** | **Test Description** | **Test Data** | **Expected Result** |
| 1 | TC-LOG-01 | Logging user into the application | **Username**:  john\_doe  **Password**:  Qwerty@123 | Should be logged in successfully |
| 2 | TC-LOG-02 | Logging user into the application with invalid credentials | **Username**:  john\_doe  **Password**:  Qwerty | Invalid user credentials error should be displayed |
| 3 | TC-LOG-03 | Logging user into the application with username that does not exist in the application | **Username**:  john\_doe123  **Password**:  Qwerty | Authentication error message should be displayed |
| 5 | TC-LOG-06 | Resetting password | **Email:**  johndoe@gmail.com | Email should be sent to the valid user with reset information |
| 6 | TC-LOG-07 | Logging in as a basic user | **Username**:  john\_doe  **Password**:  Qwerty@123 | Should be logged in successfully and should be redirected to Homepage |
| 7 | TC-LOG-08 | Logging in as an admin | **Username**:  super\_admin  **Password**:  Qwerty@123 | Should be logged in successfully and should be redirected to Admin Dashboard |

Table Test Case for Group

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SN** | **Test Case ID** | **Test Description** | **Test Data** | **Expected Result** |
| 1 | CREATE\_GROUP | Creating a new event. | **Name**: New group  **Description**: Group description here. | A new group should be created. |
| 3 | DELETE\_GROUP | Deleting an existing group. | **GROUP\_ID**: “ghjhjjkjjkljk3  7eshjd847je” | The selected group should be deleted. |
| 4 | UPDATE\_GROUP | Updating an existing group. | **GROUP\_ID**: “ghjhjjkjjkljk3  7eshjd847je”  **Name**: Group II  **Description**: Updated description. | The selected group should be updated. |
| 5 | SHOW\_ALL\_GROUPS | Viewing all the groups which you are in. |  | All the group that you joined should be visible. |
|  |  |  |  |  |

Table Test cases for Notification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SN** | **Test Case ID** | **Test Description** | **Test Data** | **Expected Result** |
| 1 | SEND\_NOTIGICATION | Send notification to the user. | Select the user to send an invitation for joining the group. | Notification should be sent to the user. |

### Unit testing and Test cases

Unit testing refers to the testing of every small modular component of the system, keeping them isolated from other modules. Here testing results of the various parts of the system were mentioned. The entire system is designed in a modularized form, and each module is tested via unit testing. The outcome of each module is checked. Every unitary module of the system is used.(2-3 figure)



Figure Backend unit testing results

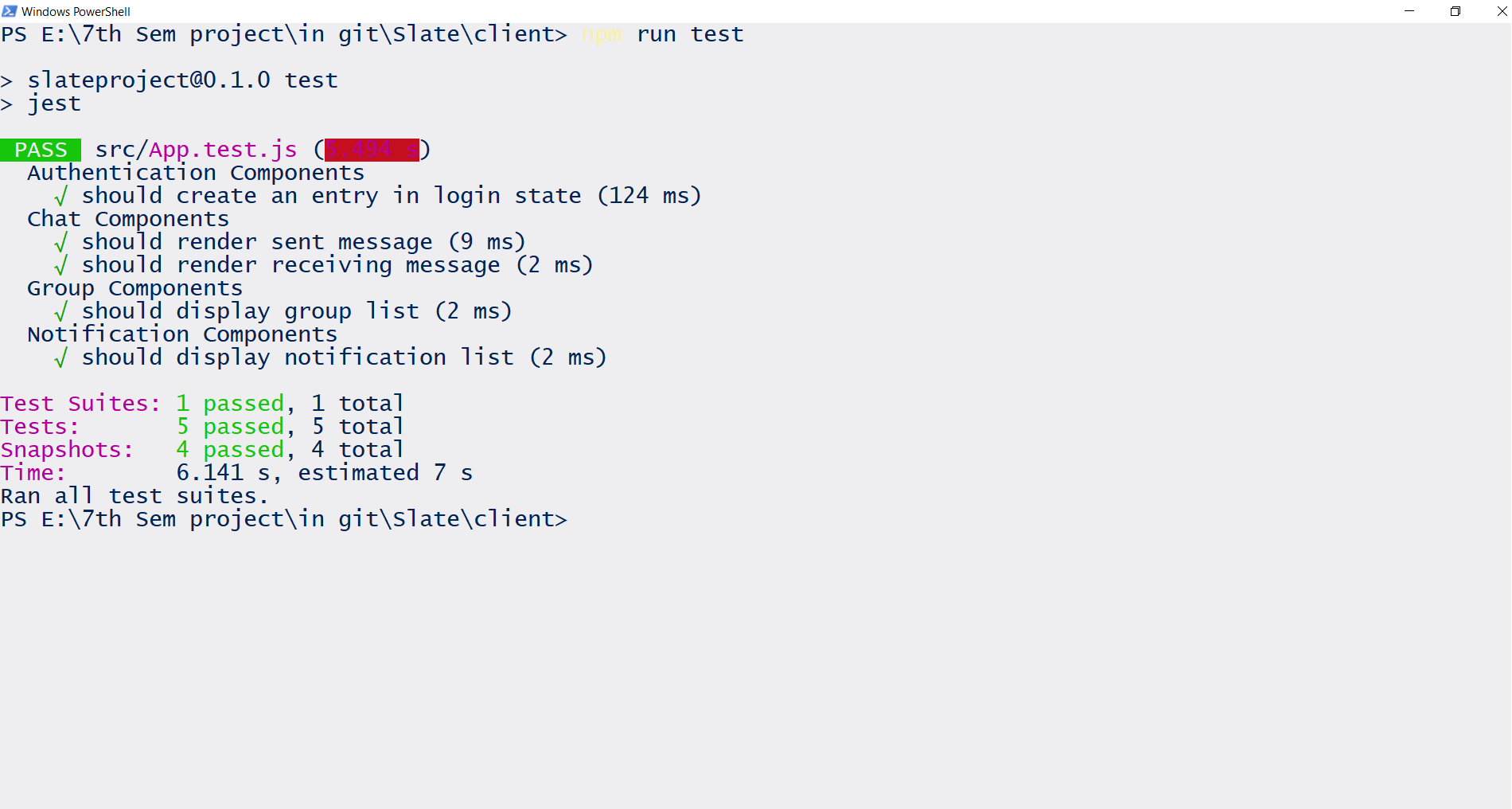


Figure 17Frontend unit testing results

# Conclusion

## Conclusion

The manual technique of planning resulted in most planners going over budget because much of the resources were focused into traveling around looking for objects and services rather than purchasing items. Traditional techniques are also inconvenient because planners must travel long distances to check on products. This is a list of functions that the system is attempting to automate. The purpose of this project was to design and develop an online event management system that would help with online event scheduling and booking. In order to attain reliability, effectiveness, and efficiency, the organization will need to use the intended system.(finding , comparison bet other system)

## Future Recommendations

The future recommendation of the project circles around maintaining information regarding:

* More advanced software for event management systems including more facilities can be given.
* The platform will be hosted on online servers to make it accessible worldwide
* Integrate multiple load balancer to distribute the loads of the system

Options were left open so any other future requirement in the system by the client for the enhancement of the system is easily implemented. In the end we would like to thank all the persons involved in the development of the system directly or indirectly.

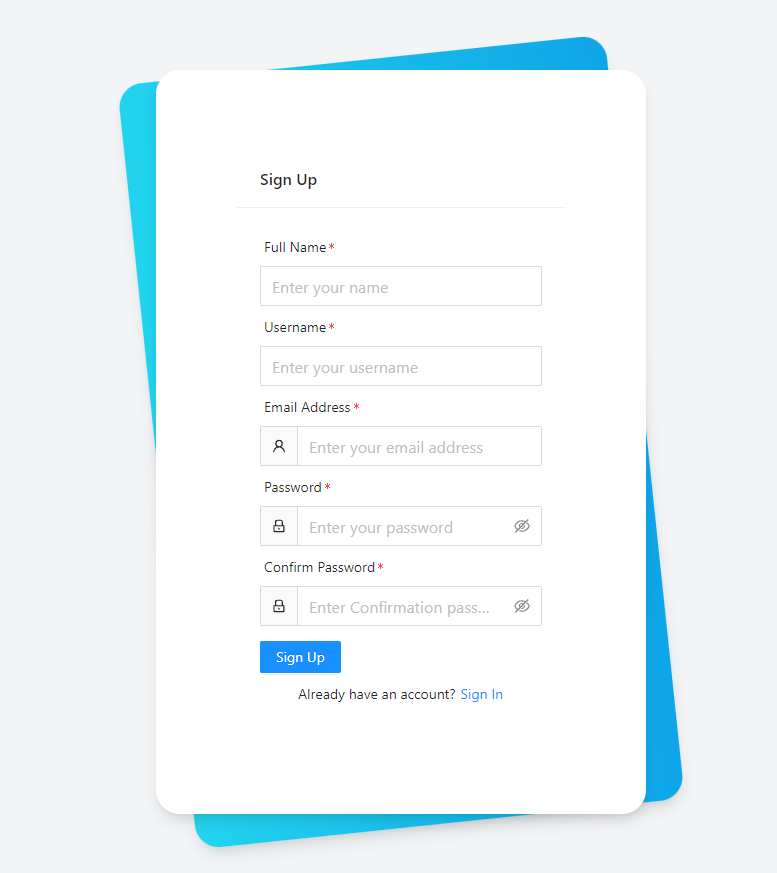
References

|  |  |  |
| --- | --- | --- |
| [1] | Huseyin Kose, Metin Argan, Mehpare Tokay Argan, "Research Gate," 2011. [Online]. Available: https://www.researchgate.net/publication/256615203\_Special\_event\_management\_and\_marketing\_A\_case\_study\_of\_TKBL\_all\_star\_2011\_in\_Turkey. [Accessed 2021]. | |
| [2] | "LocalHop," 2021. [Online]. Available: https://getlocalhop.com/a-brief-history-of-event-management-event-planning-then-and-now/ [Accessed 2021]. | |
| [3] | V. Prajapati, "Event Management System," 2017. | |
| [4] | K. McCabe, "G2," 17 May 2018. [Online]. Available: https://www.g2.com/articles/types-event-marketing. [Accessed 2021]. | |
| [5] | "10times," [Online]. Available: https://10times.com/. [Accessed 2021]. | |
| [6] | "Nevon Projects," [Online]. Available: https://nevonprojects.com/online-event-management-system/. [Accessed 2021]. | |
| [7] | Oliver Thomas, Bettina Hermes, Peter Loos, "REFERENCE MODEL-BASED EVENT MANAGEMENT". | |
| [8] | "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Event\_management. [Accessed 2021]. | |
| [9] | Eventogy, "Medium," 25 July 2019. [Online]. Available: https://medium.com/eventogy/10-things-to-remember-when-planning-a-high-profile-event-9a57f84e2879. [Accessed 2021]. | |
| [10] | | A. Kumar, "PHP Gurukul," [Online]. Available: https://phpgurukul.com/event-management-system-using-php-and-mysql/. [Accessed 2021]. |

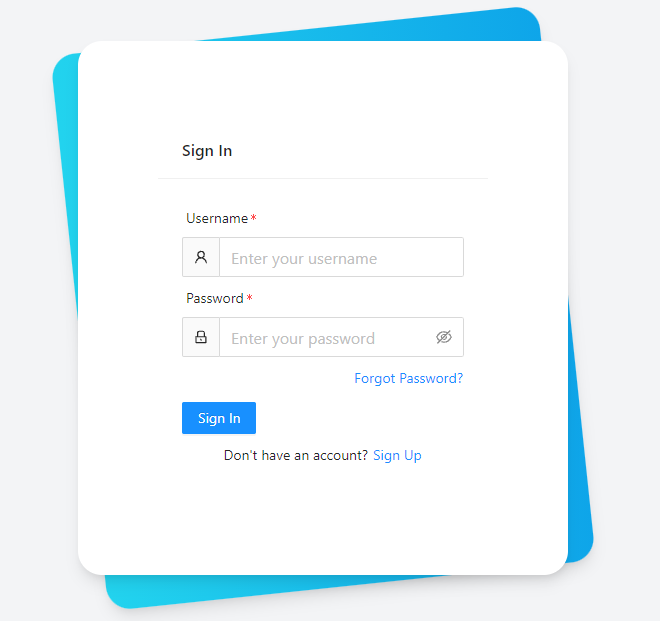
## 

## Appendix

**Sign Up and Sign in Page**

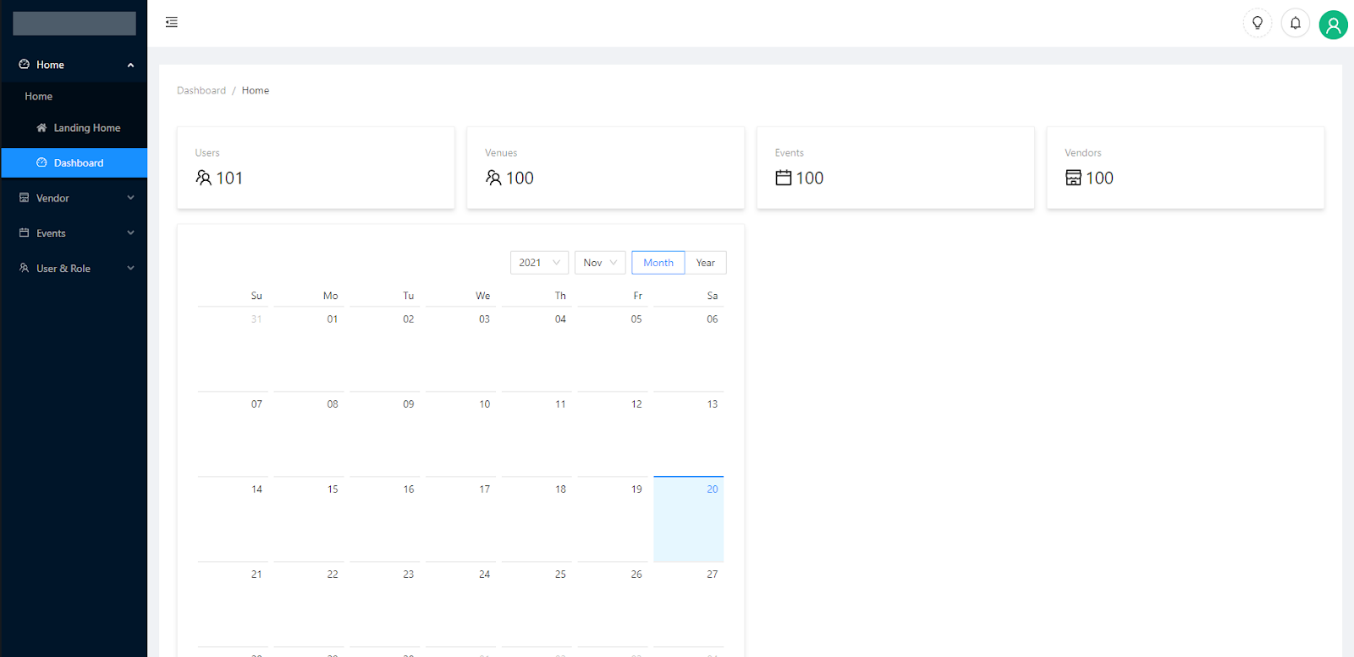


***Figure 13 Signup page***



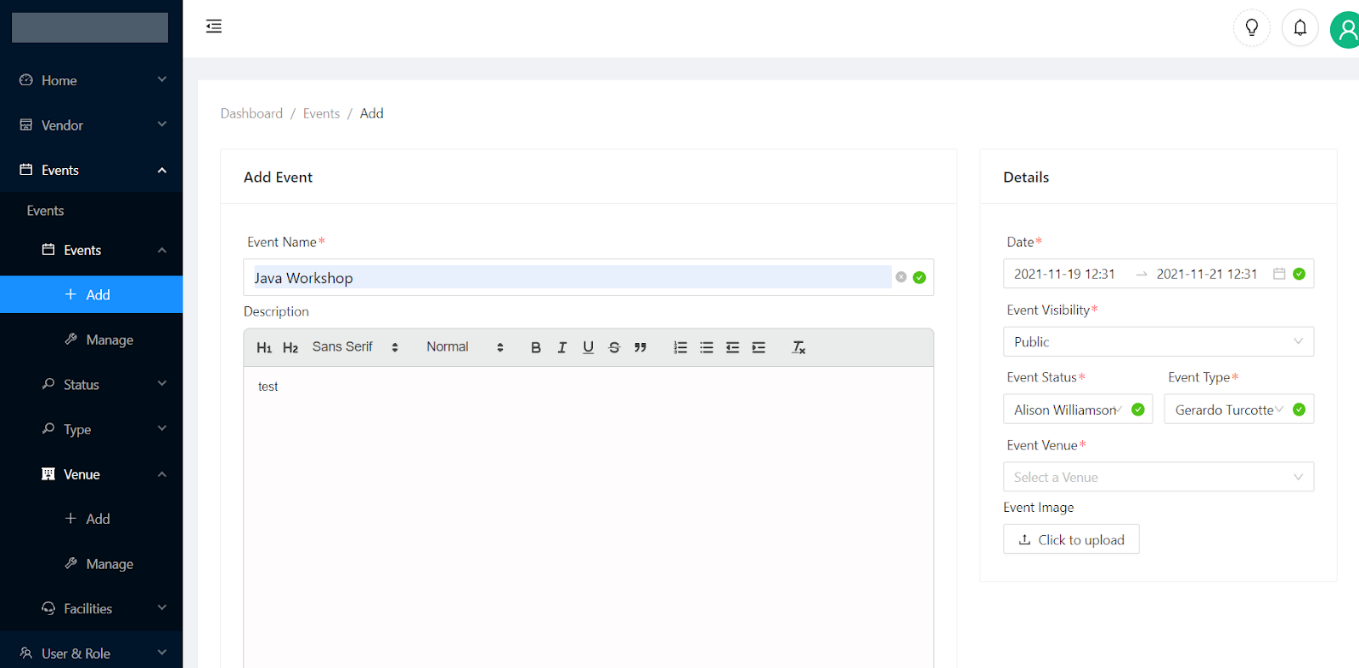
***Figure 14 Sign in Page***

**Dashboard**



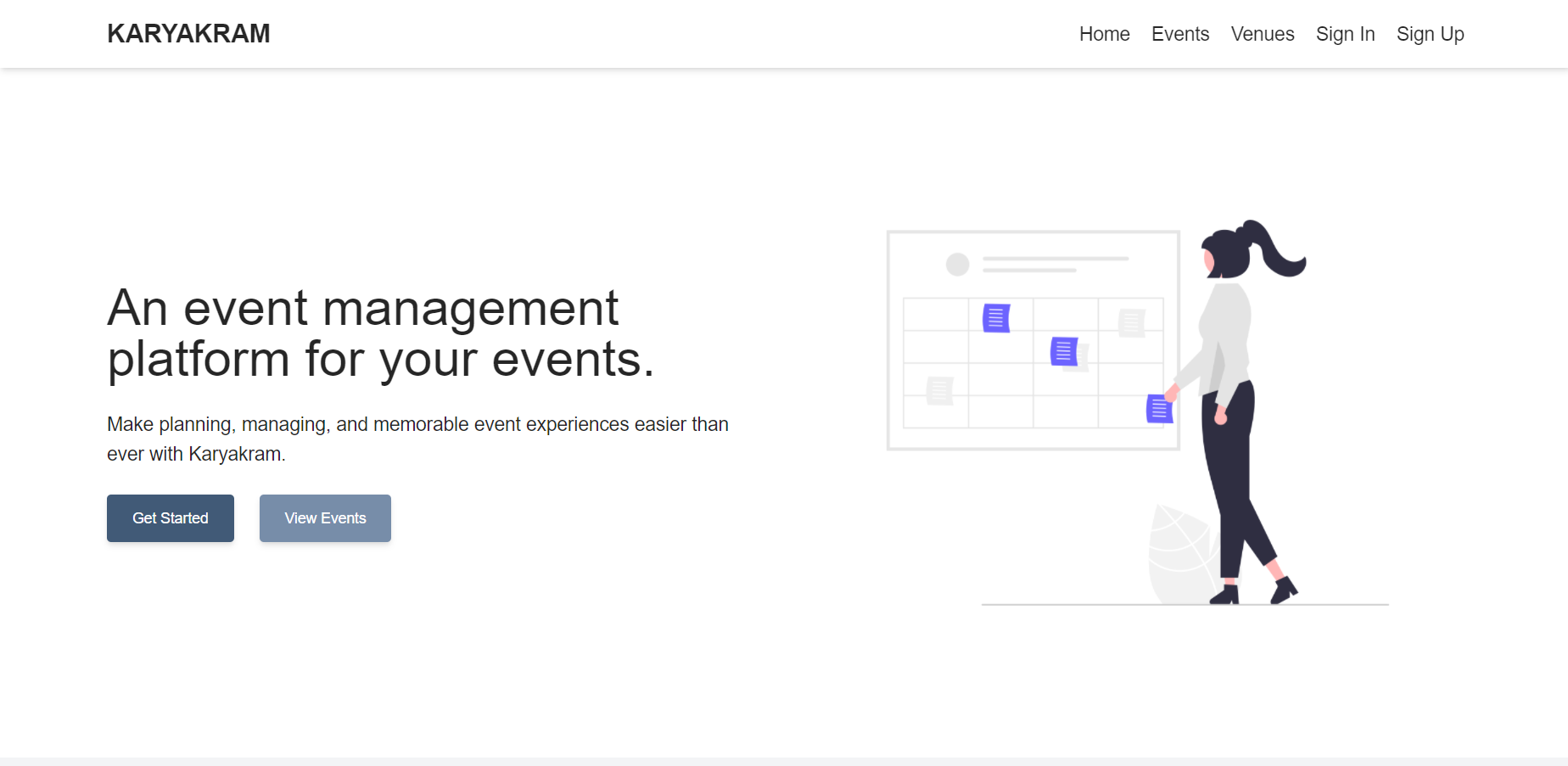
***Figure 15 Dashboard***

**Creating Group**



*Figure 17. Creating Group*

**Landing Page**



***Figure 19 Landing Page***

1. [www.zoom.org](http://www.zoom.org) [↑](#footnote-ref-1)