**STUDENT CENTRIC CLASSROOM MANAGEMENT SYSTEM**

**A Project Report**

***Submitted by:***

**WAKAS AZIZ (CSE-18-05)**

**REYAN REYAZ (CSE-18-63)**

**MINZAH RAFIQ (CSE-18-LE-69)**

***In partial fulfilment for the award of the degree***

***Of***

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**At**

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**ISLAMIC UNIVERSITY OF SCIENCE AND TECHNOLOGY**

**1-UNIVERSITY AVENUE AWANTIPORA, PULWAMA, JAMMU AND KASHMIR-192122**

**OCTOBER 2021**

# CERTIFICATE

We hereby certify that the work which has been carried out in this project entitled “ **PUBBLE - STUDENT CENTRIC CLASSROOM MANAGEMENT SYSTEM”,** in partial fulfillment of the requirement for the award of the degree of “Bachelors of Technology in Computer Science And Engineering” submitted to the Department of Computer Science and Engineering, Islamic University of Science and Technology, Awantipora, is an authentic record of our work carried out under the supervision of “**Dr Ahsan Hussain”**.

|  |  |  |
| --- | --- | --- |
| **Wakas Aziz** | **Reyan Reyaz** | **Minzah Rafiq** |
| CSE-18-05 | CSE-18-63 | CSE-18-LE-69 |

This is to certify that the statements made above by the candidates are correct and true to the best of my knowledge.

|  |  |  |
| --- | --- | --- |
| **Signature of the Guide** | **Project Coordinator** | **HOD** |

## 

**Place: Islamic University of Science and Technology**

**Date: 27-12-2021**

# ACKNOWLEDGMENT

# We take this opportunity to express our profound sense of gratitude and respect to all those who helped us throughout the duration of this project. We acknowledge the efforts of those who have contributed significantly to our project. First of all we are very thankful to Almighty for providing us such a great opportunity to do the project in this university and also very thankful to our parents for their regular support and guidance.

# We feel privileged to offer our sincere thanks and deep sense of gratitude to our HOD Dr. Asif Assad and our guide Dr. Ahsan Hussain for expressing their confidence in us by letting us work on a project of this magnitude and using latest technologies and providing their support help and encouragement in implementing this project. Last but not least, we are grateful to all our friends for providing critical feedback and support whenever required. There are times in such projects when clock beats your time and you run out of energy, you just want to finish it once and forever, however parents and friends made us endure such times with their unfailing humor and warm wishes.

Wakas Aziz

Reyan Reyaz

Minzah Rafiq

# ABSTRACT

At university, every student has a life apart from the class lectures. Students share resources, engage in discussions, important information and notices are being shared with the class all the time. Throughout our journey so far, many of us (students) very frequently miss these important information. In addition to that, from our experience, we know some classes are only relevant to only a small group of students, like the classes for re-register / backlogs or the elective subjects. Therefore collecting relevant information is not a straightforward or easy task. We as university students have always imagined a unified platform where all the dealings with the class will be nicely structured and easy to use and this is where we get the inspiration for our project. On our platform, the students will be able to join their relevant classes and all the important notices and information shared with the class will be collected in the news-feed. Each student will get their daily briefings based on their day-schedule and classes he/she has joined. And last, but not least, the students within a class will be able to engage in the class discussions through our discussion forum.

Our project is to build a classroom management system for the universities, colleges and schools, making the entire classroom experience unified, streamlined and efficient. The main features of our platform are:

* Resource sharing
* Notification and Notices
* Briefing
* Group discussions

In this project we made a prototype of this web Application with two features:

* Resource sharing

# Notification and Notices

# TABLE OF CONTENTS

|  |  |
| --- | --- |
| **CERTIFICATE** |  |
| **ACKNOWLEDGEMENT** |  |
| **ABSTRACT** |  |
| **TABLE OF CONTENT** |  |
| **CHAPTER 1:** INTRODUCTION |  |
| **CHAPTER 2:** LITERATURE REVIEW |  |
| **CHAPTER 3:** SYSTEM DESIGN AND IMPLEMENTATION |  |
| **CHAPTER 4:** RESULT |  |
| **CHAPTER 5:** CONCLUSION/RECOMMENDATION |  |
| **CHAPTER 6:** FUTURE SCOPE |  |
| **CHAPTER 7:** REFERENCES |  |
| **CHAPTER 8:** APPENDICES |  |

CHAPTER 1

**INTRODUCTION**

**Project Overview**

Classroom management system is to create a platform where the students will be able to join their relevant classes and all the important notices and information shared with the class will be collected in the news-feed. The resources shared with the class will be nicely structured and easy to access from the resources section. Each student will get their daily briefings based on their day-schedule and classes they have joined. And the students within a class will be able to engage in the class discussions through the discussion forum of our platform.

**Project Description**

This project is aimed to developing the student centric classroom management system. Our project is to build a classroom management system for the universities, colleges and schools, making the entire classroom experience unified, streamlined and efficient. The main features of our platform are:

* Resource sharing
* Notification and Notices
* Briefing
* Outline

Currently we are using multiple applications/platforms to manage the dealings with the class. There platforms are not tailored for managing the classroom and therefore managing and keeping track of all these platforms is very difficult. Through our classroom management web-application, our goal is to streamline and unify the dealings with the class by eliminating the need for multiple platforms for sharing resources, important information and engage in group discussions and get daily briefings about the relevant classes. These difficulties and problems in our daily classroom life is what synthesized the main modules of our platform. The objective of our application is to manage the information sharing, resource sharing, daily briefings and group discussions. The project is mainly targeted to university and college students making their interaction with the class easier.

**Our features:**

* **News-feed:** where all the important notices and resources posted in the classroom will be collected.
* **Resources page:** Where all the resources shared with the class will appear. Resources can also be made available globally throughout the platform and users will be able to search through them.
* **Dynamic briefings:**  will bring daily briefings tailored to each student for their daily schedule and plan.
* **Discussion forum:** where all students of a class can chat with the class.

CHAPTER 2

**LITERATURE REVIEW**

**Moodle:**

Moodle is a free and open-source learning management system (LMS) written in PHP and distributed under the GNU General Public License. Developed on pedagogical principles, Moodle is used for blended learning, distance education, flipped classroom and other e-learning projects in schools, universities, workplaces and other sectors. With customizable management features, Moodle used to create private websites with online courses for educators and trainers to achieve learning goals. Moodle allows for extending and tailoring learning environments using community-sourced plugins.

**PIAZZA:**

Piazza is a learning management system which allows students to ask questions in a forum-type format. Instructors are able to moderate the discussion, along with endorsing accurate answers. The software was invented by Pooja Nath in 2009 in order to speed response times and create a common place where students could engage in discussion outside of the classroom. Utilizing an extensive notification system and a simple layout, the response time on piazza averages approximately 14 minutes. Instructors also have the ability to allow students to post anonymously, encouraging more in-depth discussion. The word piazza comes from the Italian word for plaza- a common city square where people can come together to share ideas and knowledge.

CHAPTER 3

**SYSTEM DESIGN &**

**IMPLEMENTATION**

Our platform is based on the **client-server** architecture. The client and the server are decoupled and stand-alone applications. The server constitutes of a REST API on which the client makes requests and is served the data.

For any modern dynamic application, there can be one of two possibilities:

* It can be either a client-side-rendered application, or
* It can be a server-side-rendered application

For this application, we choose the client-side-rendered architecture. The reason behind this was to keep the server simple and separate from the presentational logic. The data between the client and the server is shared in the JSON format.

The entire application is broadly split into two parts:

* The front-end, and
* The back-end

**FRONT-END**

Under front-end, we designed and developed the view of the prototype. The technologies used in front-end are:

* HTML : to define the structure (skeleton) of the website.
* CSS : to style the webpage.
* Javascript : to add interactivity and make dynamic requests to the API on the server.

The front end consists of a web application which sends the request for the notifications or resources to the server. The server responds with the data in the JSON format and the application parses the data and add the elements to the UI. As this is a dynamic web application, the content to be displayed depend on the documents stored in the database.

**HTML:-**

**HTML** (HyperText Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. Other technologies besides HTML are generally used to describe a web page's appearance/presentation (CSS) or functionality/behavior (JavaScript).

"Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web.

HTML uses "markup" to annotate text, images, and other content for display in a Web browser. HTML markup includes special "elements" such as <head>, <title>, <body>, <header>, <footer>, <article>, <section>, <p>, <div>, <span>, [<img>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/img), <aside>, <audio>, <canvas>, [<datalist>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/datalist), <details>, <embed>, [<nav>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/nav), <output>, <progress>, <video>, [<ul>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ul), [<ol>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ol), <li> and many others.

An HTML element is set off from other text in a document by "tags", which consist of the element name surrounded by "<" and ">".  The name of an element inside a tag is case insensitive. That is, it can be written in uppercase, lowercase, or a mixture. For example, the <title> tag can be written as <Title>, <TITLE>, or in any other way.

**CSS:-**

**Cascading Style Sheets** (**CSS**) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media. CSS is among the core languages of the **open web** and is standardized across Web browsers according to W3C specifications.

CSS is used to style and layout web pages — for example, to alter the font, color, size, and spacing of your content, split it into multiple columns, or add animations and other decorative features.

**JAVA SCRIPT:-**

**JavaScript** (**JS**) is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat. JavaScript is a multi-paradigm, dynamic language with types and operators, standard built-in objects, and methods. Its syntax is based on the Java and C languages — many structures from those languages apply to JavaScript as well. JavaScript supports object-oriented programming with object prototypes, instead of classes. JavaScript also supports functional programming — because they are objects, functions may be stored in variables and passed around like any other object.

JavaScript programs manipulate values, and those values all belong to a type. JavaScript's types are:

* Number
* [BigInt](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Data_structures#bigint_type)
* String
* Boolean
* Symbol (new in ES2015)
* Object
  + Function
  + Array
  + Date
  + [RegExp](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/RegExp)
* null
* undefined

And there are some built-in Error types as well.

**BACK END**

Under this part of the application, we designed and developed the REST API for project, connecting the server app with the database. Technologies used in the back-end are:

* Node.js
* Express
* **MongoDB**

In this mini project, we made a prototype of our application implementing two main features:

* Resources
* Notifications and Notices

On the application level, the prototype has three models:

* User model : Necessary for implementing authentication and Authorization
* Resources model : For implementing resources uploading and download functionality
* Notification model : For uploading and retrieving notifications and notices

To bring functionality out of the models and implementing other application specific functionality, we have programmed different controllers. The controllers that we developed are:

* Authentication controller : for implementing authentication and authorization to our application.
* Error controller : to handle all the predicted errors and exceptions that can happen in the runtime.
* Notification controller : to enable all the CRUD operations on the notifications.
* Resource controller : to enable all the CRUD operations on the resources.
* User controller : to enable CRUD operation on the user. These crud operations are mainly restricted to the admin of the application.

**NODE JS:-**

Node (or more formally *Node.js*) is an open-source, cross-platform runtime environment that allows developers to create all kinds of server-side tools and applications in JavaScript. The runtime is intended for use outside of a browser context (i.e. running directly on a computer or server OS). As such, the environment omits browser-specific JavaScript APIs and adds support for more traditional OS APIs including HTTP and file system libraries.

From a web server development perspective Node has a number of benefits:

* Great performance! Node was designed to optimize throughput and scalability in web applications and is a good solution for many common web-development problems (e.g. real-time web applications).
* Code is written in "plain old JavaScript", which means that less time is spent dealing with "context shift" between languages when you're writing both client-side and server-side code.
* JavaScript is a relatively new programming language and benefits from improvements in language design when compared to other traditional web-server languages (e.g. Python, PHP, etc.) Many other new and popular languages compile/convert into JavaScript so you can also use TypeScript, CoffeeScript, ClojureScript, Scala, LiveScript, etc.
* The node package manager (NPM) provides access to hundreds of thousands of reusable packages. It also has best-in-class dependency resolution and can also be used to automate most of the build toolchain.
* Node.js is portable. It is available on Microsoft Windows, macOS, Linux, Solaris, FreeBSD, OpenBSD, WebOS, and NonStop OS. Furthermore, it is well-supported by many web hosting providers, that often provide specific infrastructure and documentation for hosting Node sites.
* It has a very active third party ecosystem and developer community, with lots of people who are willing to help.

We can use Node.js to create a simple web server using the Node HTTP package.

## Node Package Manager (npm)

[npm](https://www.npmjs.com/) is bundled with Node.js. It runs on the command line as the command npm. npm is committed to making JavaScript development elegant, productive, and safe. The free npm Registry has become the center of JavaScript code sharing, and with more than one million packages, the largest software registry in the world. It is a package manager that downloads packages into a node\_modules folder. We call the downloaded packages through const libraryModule = require("libraryname").

## EXPRESS:-

## Express is a popular unopinionated web framework, written in JavaScript and hosted within the Node.js runtime environment. Express is the most popular Node web framework, and is the underlying library for a number of other popular Node web frameworks. It provides mechanisms to:

* Write handlers for requests with different HTTP verbs at different URL paths (routes).
* Integrate with "view" rendering engines in order to generate responses by inserting data into templates.
* Set common web application settings like the port to use for connecting, and the location of templates that are used for rendering the response.
* Add additional request processing "middleware" at any point within the request handling pipeline.

While Express itself is fairly minimalist, developers have created compatible middleware packages to address almost any web development problem. There are libraries to work with cookies, sessions, user logins, URL parameters, POST data, security headers, and many more.

**MongoDB:-**

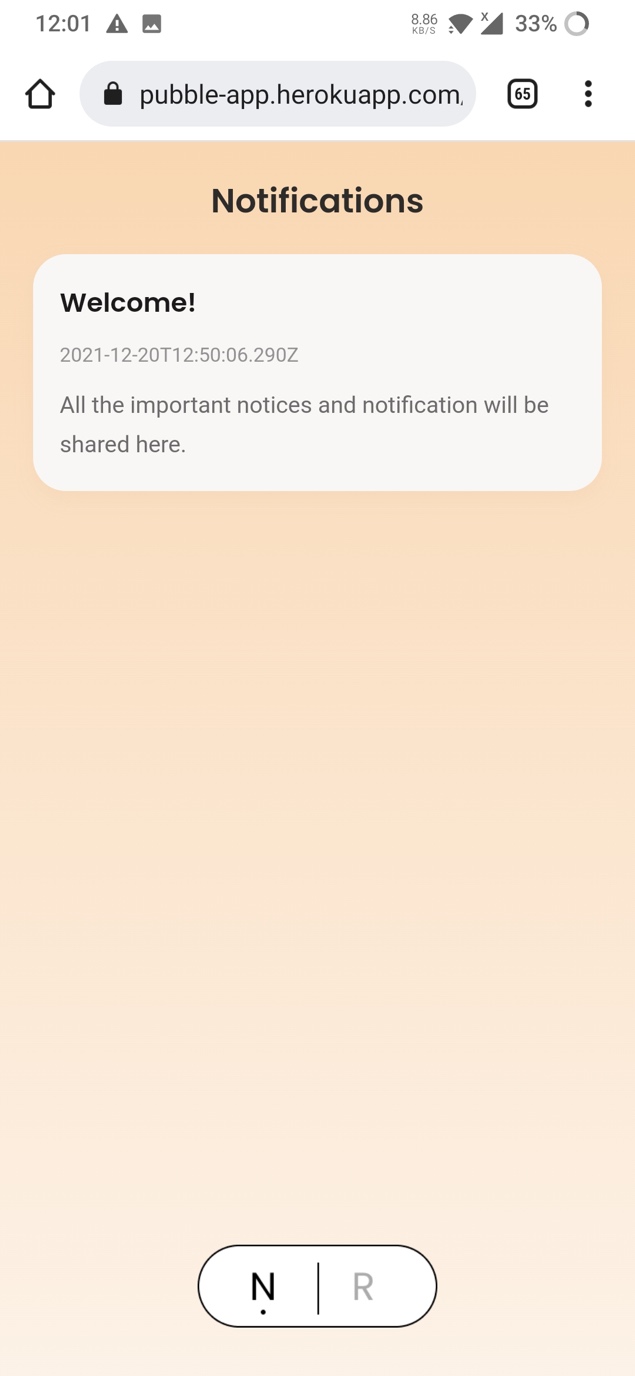
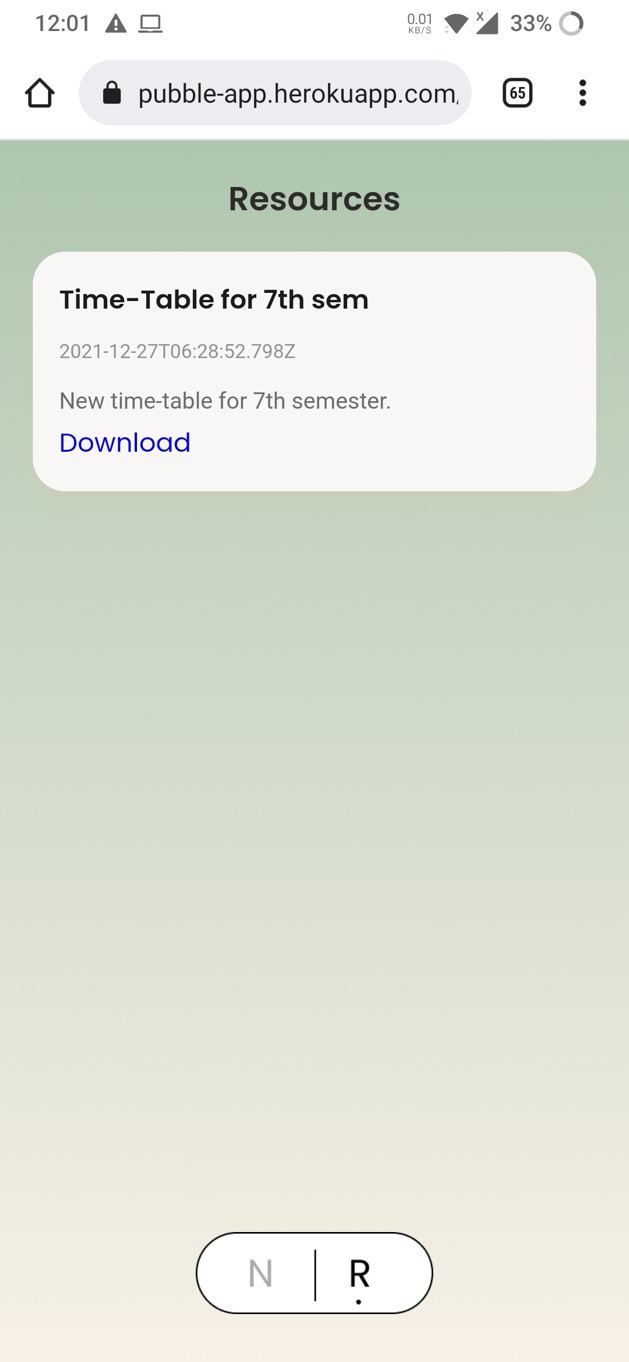
* MongoDB is a document database with the scalability and flexibility that you want with the querying and indexing that we need. MongoDB stores data in flexible, JSON-like documents, meaning fields can vary from document to document and data structure can be changed over time
* The document model maps to the objects in your application code, making data easy to work with
* Ad hoc queries, indexing, and real time aggregation provide powerful ways to access and analyze our data
* MongoDB is a distributed database at its core, so high availability, horizontal scaling, and geographic distribution are built in and easy to use
* MongoDB is free to use.

CHAPTER 4

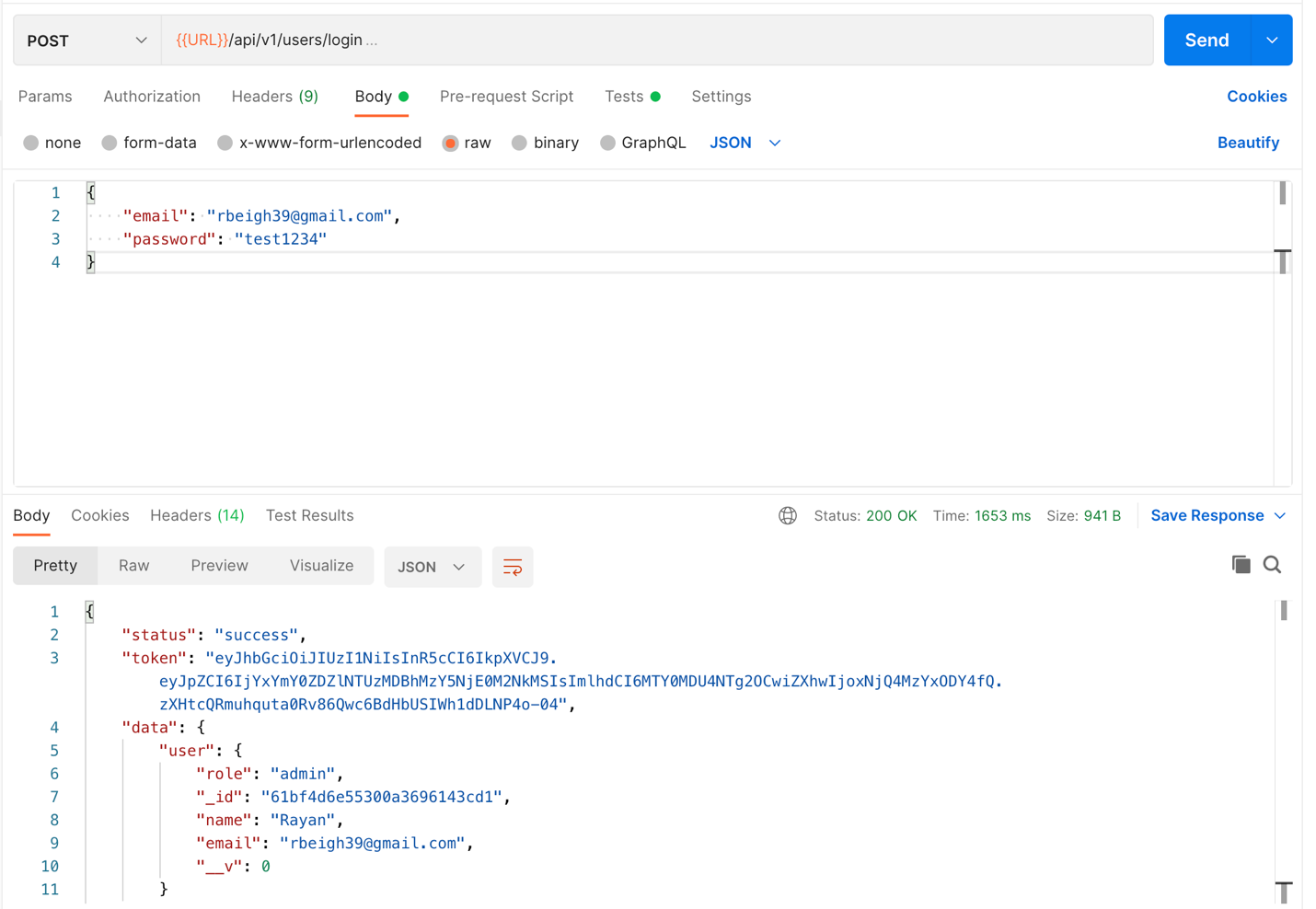
**RESULTS AND OUTPUTS**

**UI**

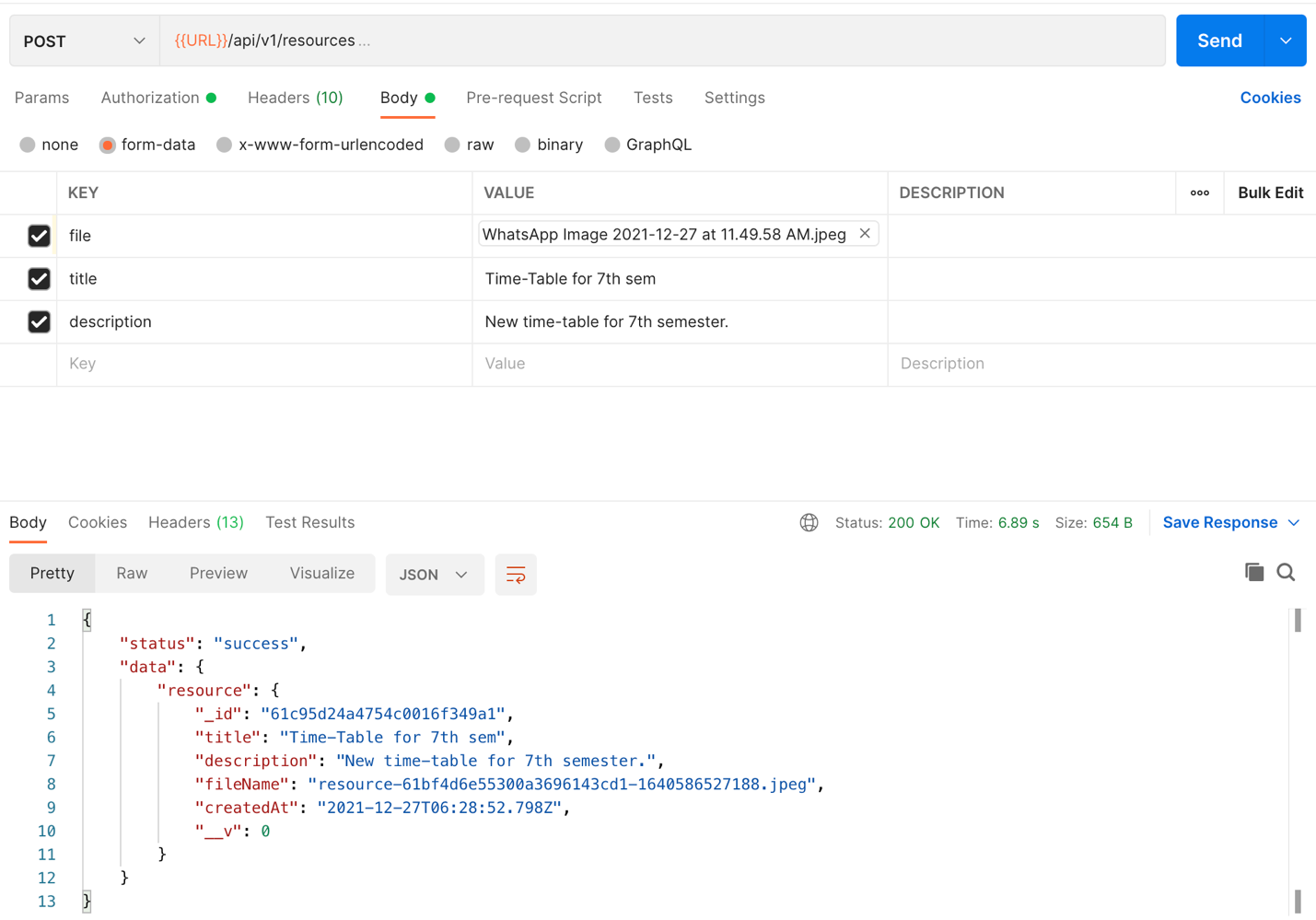
Below are the screenshots of the interface for receiving resources and notifications.



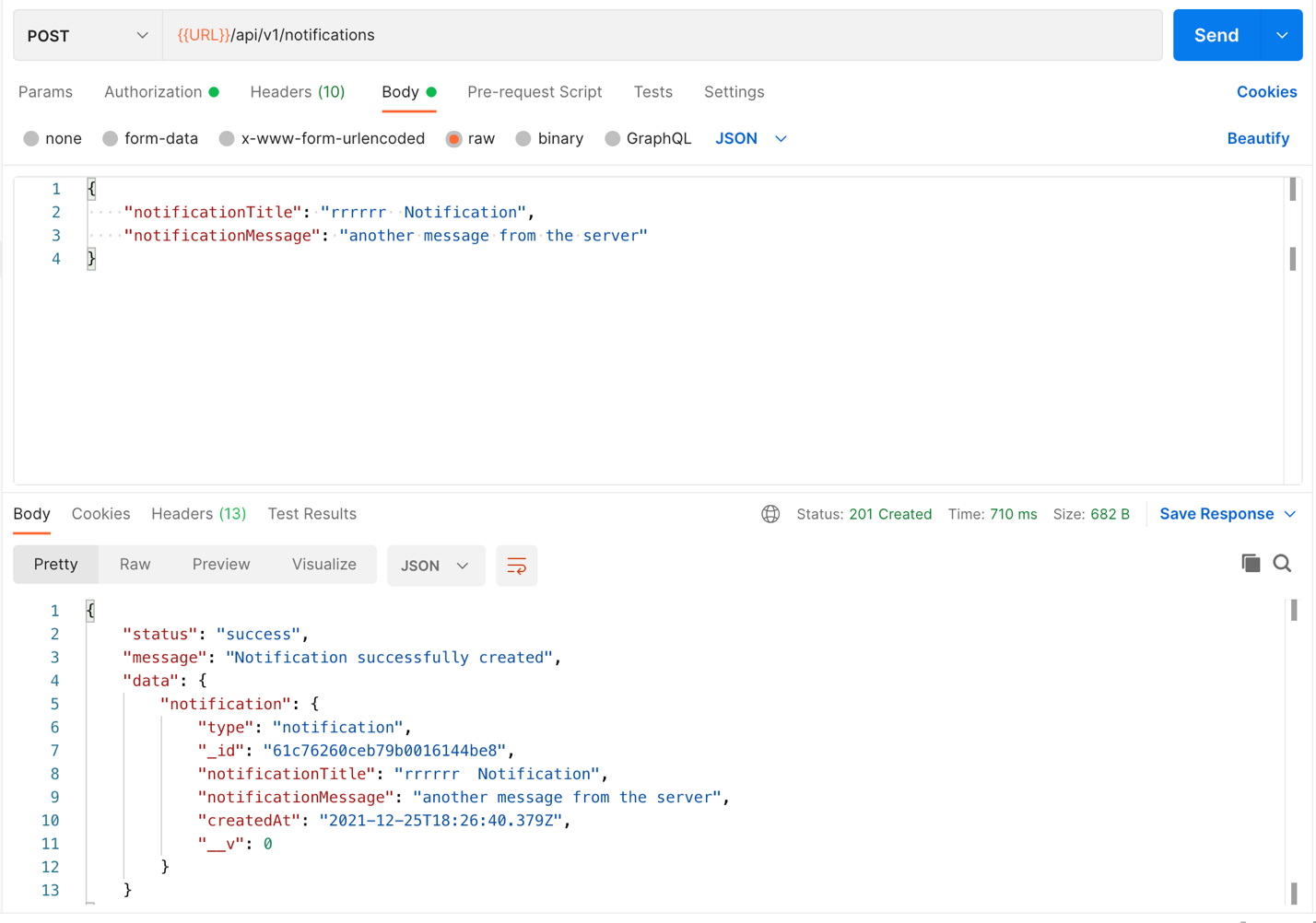
Following are the screenshots of logging in, creating new resource and creating new notification:



*Login request from postman*



*POST request for making a new resource in postman.*



*POST request for creating a new notification/notice*

CHAPTER 5

**CONCLUSION AND RECOMMENDATION**

In this project we made a prototype of this web application with two features

* Resource Sharing
* Notification and Notices

The resulting application was deployed on Heroku and be accessed on <https://pubble-app.herokuapp.com/notifications.html>

It was recommended to us to include online-courses functionality in out platform to create a truly unified system for managing classroom.

It was also recommended to also include teachers in out platform and develop some sort of functionality to enhance student-teacher interaction.

CHAPTER 6

**FUTURE SCOPE**

The future scope of the project are as follows:

* Developing a more powerful search engine including implementing the relevance scoring.
* Designing and developing all the functionalities of this application.
* Develop a system for online courses within our platform.
* Briefing system, which will be like an agent which will brief the students about their daily schedule. The briefings will be unique to each student based on the courses/classes he/she has joined.

**REFERENCES**

[1] HTML.com, HTML, <https://html.com/#ixzz7GAfJTjGC>

[2] CSS, MDN Web Docs https://developer.mozilla.org/en-US/docs/Learn/CSS

[3] JavaScript, MDN Web Docs

[4] Nodejs Express, MDN Web Docs, https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express\_Nodejs

[5] MongoDb, https://www.mongodb.com/what-is-mongodb

[6] Moodle, <https://moodle.org/>

[7] Piazza, <https://piazza.com/>

Research Papers:

[8] A Review of Research on Classroom Management in Turkey: Mehmet Erdogana, Abdulkadir Kurt: *5th World Conference on Learning, Teaching and Educational Leadership, WCLTA 2014:*

<https://www.researchgate.net/publication/277971330_A_Review_of_Research_on_Classroom_Management_in_Turkey>

[9] Towards an understanding of social media use in the classroom: Antoine Van Den Beemt:

<https://www.tandfonline.com/doi/ref/10.1080/1475939x.2019.1695657?scroll=top>

[10] A Qualitative Study on Classroom Management and Classroom Discipline Problem, Reasons, and Solutions: A Case of Information Technologies Class: Mehmet Erdogan, Engin Kursun, Engin Kursun, Gulcin Tan Sisman, Faith Saltan, Ali Gok, Ismail Yildiz:

*Educational Sciences: Theory and Practice 10 (2), 881-891, 2010.*