A

CAPSTONE PROJECT

REPORT ON

"Do Connect"

Submitted by: -

CAPSTONE GROUP G1

TEAM MEMBERS

DASARI SUMANTH

RAMYASRI PACHHIGOLLA

CHAKRADHAR JAKKAMSETTI

PRATHAMESH MAHENDRA GARDE

RAJESH VILLA

Under the Guidance of

Mr. JAVEED MOHAMMED HUSNUDDIN

ACKNOWLEDGEMENT

we take this opportunity to thank all those who have contributed to the successful completion of a capstone project. We sincerely wish to express our gratitude to mentor Mr. JAVEED MOHAMMED HUSNUDDIN for full support, expert guidance, and encouragement and kind cooperation throughout the course. We are greatly indebted to him for his help throughout project work. We express our sincere gratitude towards Ms. Anisha ma'am, from Great Learning for providing necessary facilities, guidance and support. We would like to express our gratitude towards our parents & our friends for their kind, co-operation and encouragement which help us a lot in completing this project. Our thanks and appreciations also go to our colleague in developing the project. Thank you to all the people who have willingly helped us out with their abilities.

LIST OF CONTESTS

ABSTRACT

Platforms for questions and answers (QA) make use of the strength of online communities to find answers or learn more. While these websites give essential information, it is important that the data be of a good standard and that users can rely on the data. Given the widespread usage of software in all facets of modern society, this is especially important for software development. The Question Answering (QA) websites, which provide many questions and answers submitted by QA users to provide rich data sources not available on web search engines and QA websites, have become quite well-known. Web users can search QA websites for solutions to their questions, but they frequently must (I) wait for quite a long time period before other QA users respond to their critical questions with even incorrect, annoying, or spam replies, or (ii) responses and constrained answer sets produced by QA websites as a result of the precise proximity used and established between archival questions and user-generated questions. We are introducing a QA separate programme called Q AR in response to critical enhancements in obtaining top-notch responses to client Q questions on the QA website. At QAR, you find several questions that are similar to Q or Q-based on its conclusions. When choosing responses for Q substantial level questions in QAR, different ratings for likenesses and answer length are considered.

PROBLEM STATEMENT

Do Connect is a popular Q and A form in which techniques questions was asked and answer.

FUNCTIONAL REQUIREMENTS:

User Stories:

1. As a user I should be able to login, Logout and Register into the application.
2. As a user I should be able to ask any question under any topic.
3. As a user I should be able to search the question on any string written in search box.
4. As a user I should be able to Answer any question asked.
5. As a user I should be able to answer more than one question and more than one time.
6. As a user I should be able to chat with other users.
7. As a user I should be able to upload images to refer.

Admin Stories:

1. As an Admin I should be able to login, Logout and Register into the application.
2. As an Admin I should be able to get mail as soon as any new Question is asked, or any Answers given.
3. As an Admin I should be able to approve the question and Answer. Any Question or Answer will be visible on the platform only if it is approved.
4. As an Admin I should be able to delete inappropriate Questions or Answers.

Software Requirements:

- **Technologies and Languages Technologies:**
    1. Angular
    2. Spring Boot
- **IDE:**
    1. Visual Studio Code
    2. Spring Tool Suite
    3. MySQL Workbench
    4. Eclipse
- **Languages:**
    1. TypeScript
    2. Java
    3. SQL

Hardware Requirements:

1. Operating System: Windows 7/8/10/11 OR MacOS X.
2. Processor: Intel or AMD dual core x86 processor.
3. RAM: 2GB or more.
4. Hard disk: 500MB free space or more.

INTRODUCTION

Do Connect is a website where programmers may ask and answer questions. It offers questions and answers on many different computer programming-related topics. In contrast to earlier question-and-answer websites like Experts-Exchange knowledge on the Unknown Things, it was designed to be a more accessible option. Stack overflow resembles a cross between Wikipedia and programming Reddit, but without the revolting sleaze and legal nevertheless unauthorized search engine gaming. It is created by programmers, for programmers, with the ultimate goal of enhancing the global body of sound programming knowledge. regardless of the programming language you employ or the operating system you prefer. Our aim is better programming. It is designed to facilitate question-and-answer sessions, being helpful. The youthful generation will have access to resources in the future that are more like to Wiki than they do today.

ANGULAR ARCHITECTURE

A platform or framework called Angular is used to create client-side apps in TypeScript and HTML. It has TypeScript code. As a collection of TypeScript libraries that are loaded into applications, it implements both core and optional functionality.

There are eight main blocks of Angular:

1. Module
2. Component
3. Metadata
4. Template
5. Data Binding
6. Service
7. Directive
8. Dependency Injection

Module:

Angular modules, often known as Ng Modules, are Angular's proprietary framework for modularity in apps. Every Angular app has at least one Angular module class, commonly known as App Module, which is the root module. A small application might simply have a root module, but most apps include many more feature modules, each of which is a well-organized chunk of code devoted to a particular application domain, workflow, or group of features.

A single metadata object whose properties describe the module is passed to the decorator function Ng Module. The following characteristics are important:

- **Declarations** – which view classes are a part of this module. The three different types of view classes in Angular are components, directives, and pipes.
- **Exports -** the subset of declarations that ought to be accessible and used in other modules' component templates
- **Imports -** Other modules with exported classes that are required by this module's defined component templates.
- **Providers -** Service providers who contributed to this module's worldwide collection of services can now access their services from anywhere in the app.
- **Bootstrap -** the root component, or primary application view, which houses all other application views. This bootstrap property should only be set by the root module.

Component:

The most fundamental UI building block in an Angular app is a component. A tree of Angular components can be found in an Angular app. A subset of directives known as angular components are always linked to a template. In contrast to other directives, a template element can only have one instantiated component at a time.

Metadata:

In order to configure a class anticipated behaviour, metadata is used to adorn the class. The different components of metadata are listed below. These are class-level decorators known as annotations. This array serves as both a @Component and @Routes decorator example.

Template:

An HTML template instructs Angular on how to render a component. Views are often set up in a hierarchical manner, enabling you to change, display, or conceal entire UI parts or pages at once. The host view of a component is specified by the template that is directly linked to it.

Data Binding:

In AngularJS, data binding refers to the synchronisation of the model and the view. When data in the model is updated, the view is updated to reflect the change, and vice versa when data in the view is modified.

Service:

Any value, feature, or function that your application requires falls under the wide category of service. A service can be almost anything. A service is often a class with a specific, focused objective. It must perform a given task competently.

Examples include:

- Message bus
- Tax calculator
- Logging service
- Data service
- Application configuration

Services don't have any Angular-specific features. A service is not defined in Angular. There is no base class for services, and there is nowhere to register services. Services, however, are essential to every Angular application. Large users of services are components.

Directive:

Directives are classes that give the template's elements new behaviour or change their current behaviour. In essence, directives are used to modify the DOM, such as by adding or removing elements from it or altering their visual appearance.

Dependency Injection:

Using the design pattern known as dependency injection, or DI, a class requests dependencies from outside sources as opposed to constructing them. Upon instantiation, Angular's DI framework offers dependencies to a class. To make your applications more adaptable and modular, use Angular DI.

SPRINGBOOT ARCHITECTURE

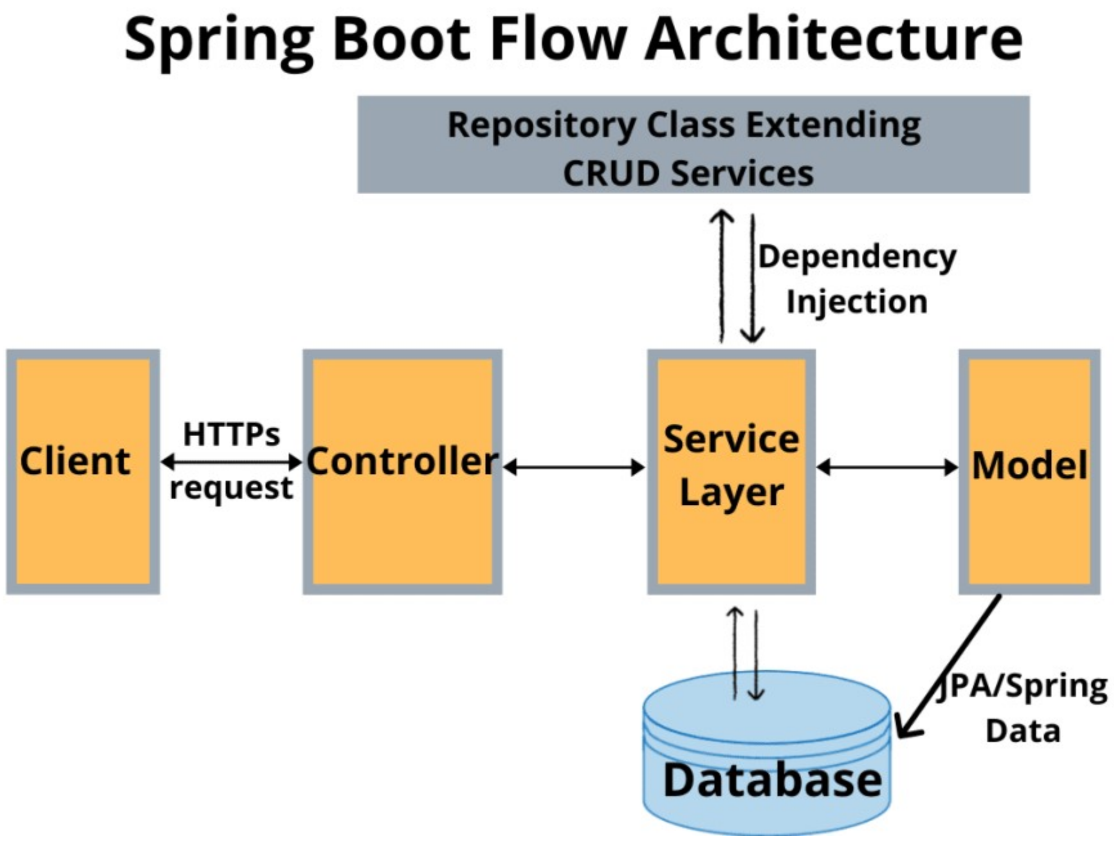


Fig. 1 Springboot Architecture

The spring boot consists of the following four layers:

- **Presentation Layer:** Authentication & Json Translation.
- **Business Layer:** Business Logic, Validation & Authorization.
- **Persistence Layer:** Storage Logic.
- **Database Layer:** Actual Database.

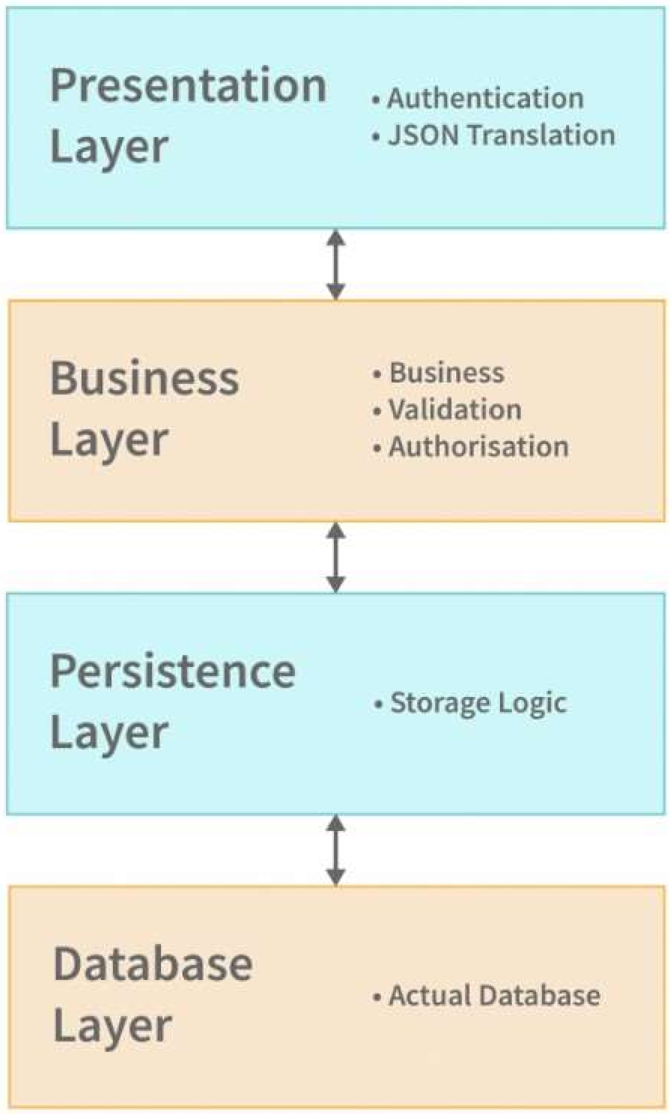


Fig. 2 Springboot Layers

Presentation Layer:

The top layer of the spring boot architecture is the presentation layer. It comprises of Views, or the application's front-end. Both HTTP requests and authentication are handled by it.

Persistence Layer:

The storage logic is entirely contained in the persistence layer, which also converts business objects into and out of database rows. Database Layer: CRUD (create, retrieve, update, delete) actions are carried out at the database layer.

Database Layer:

All databases, including MySQL, MongoDB, and others, are part of the database layer. This layer may house several databases. It is in charge of carrying out CRUD tasks.

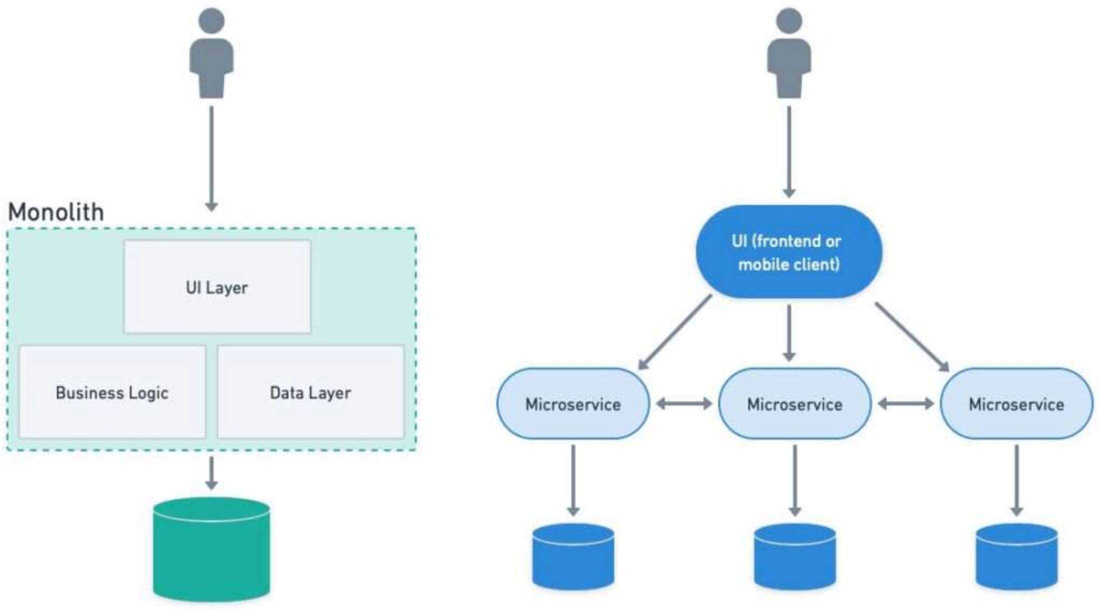


Fig. 3 Database Architecture

- Micro services are frequently used to hasten the development of applications.
- Java-based microservice architectures, particularly those based on Spring Boot, are widespread.

- The term "microservices architecture" refers to an architectural design approach for creating applications. Using microservices, a huge programme can be divided into smaller, independent components, each with an own set of responsibilities.
- Follow the three C's of microservices: componentize, cooperate, and connect when you're ready to start implementing a microservices architecture and the related development and deployment best practises.

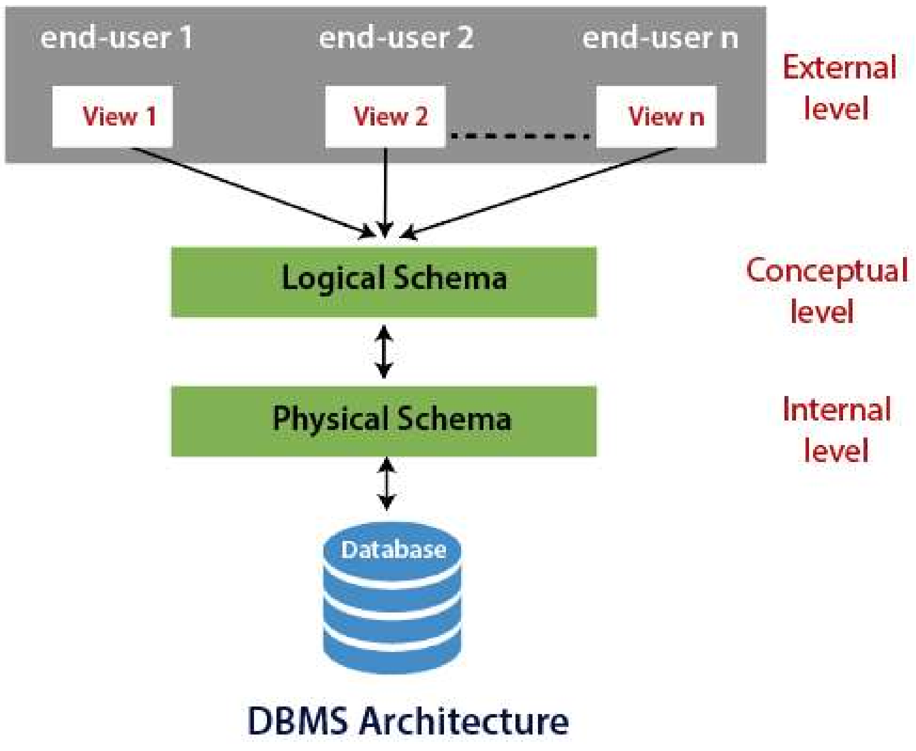


Fig. 4 Database Architecture

- A DBMS design is represented by a database architecture. It aids in the creation, growth, use, and maintenance of the database management system. The database system can be divided into separate parts that can be independently modified, changed, replaced, and altered thanks to DBMS design.
- The Database Management System (DBMS) architecture demonstrates how users interact with database data. It is unconcerned with how the

DBMS manages and processes the data. It aids in the creation, upkeep, and deployment of a database that stores and arranges information for businesses.

Entities in the Database include:

- Admin
- User
- Question
- Answers
- Image model

PROJECT OVERVIEW DIAGRAM

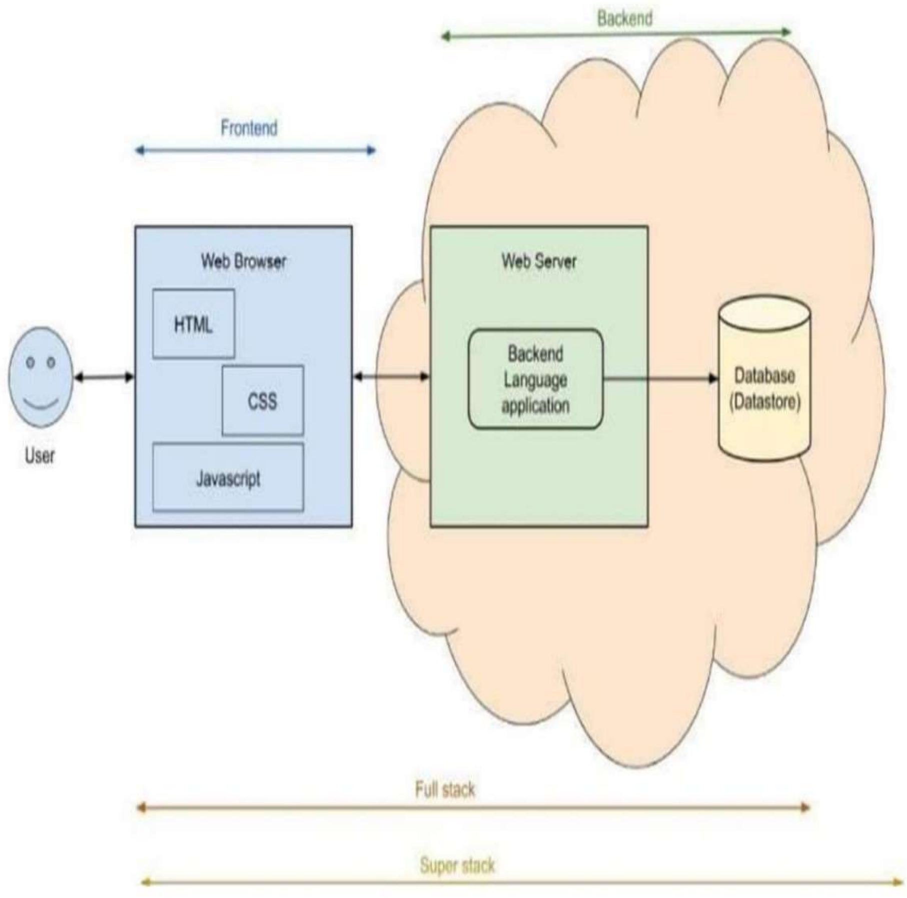


Fig No.5 project overview diagram

USE CASE DIAGRAM

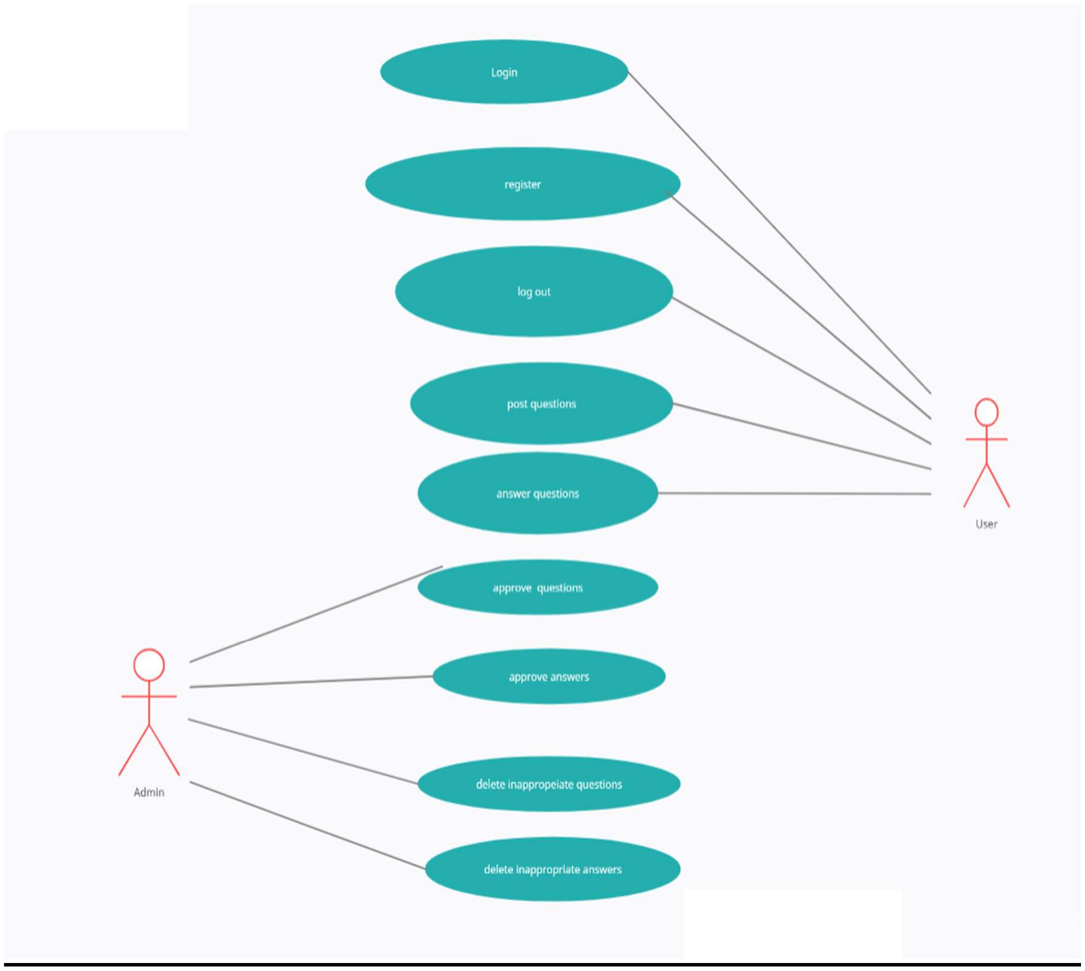


Fig No.6 Use case diagram

CLASS DIAGRAM

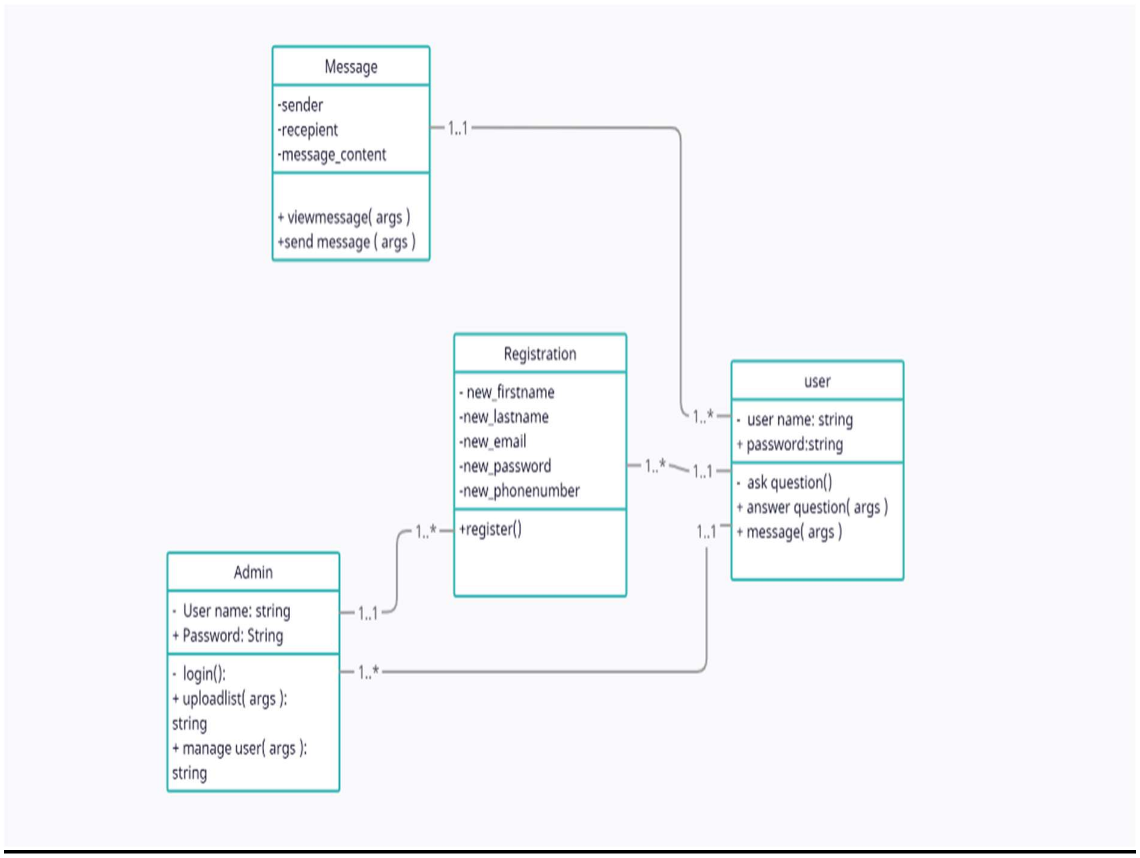


Fig No.7 Class Diagram

USER SEQUENCE DIAGRAM

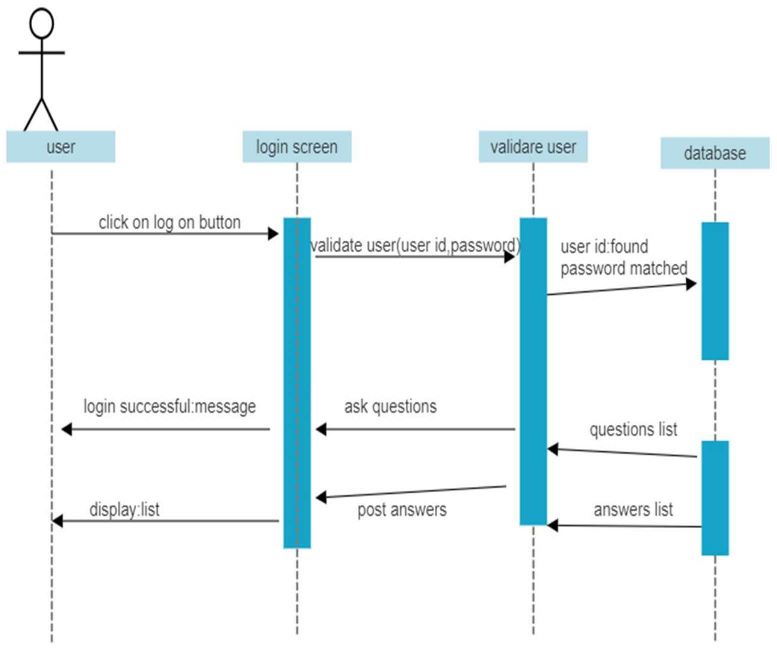


Fig No.8 User Sequence Diagram

ADMIN SEQUENCE DIAGRAM

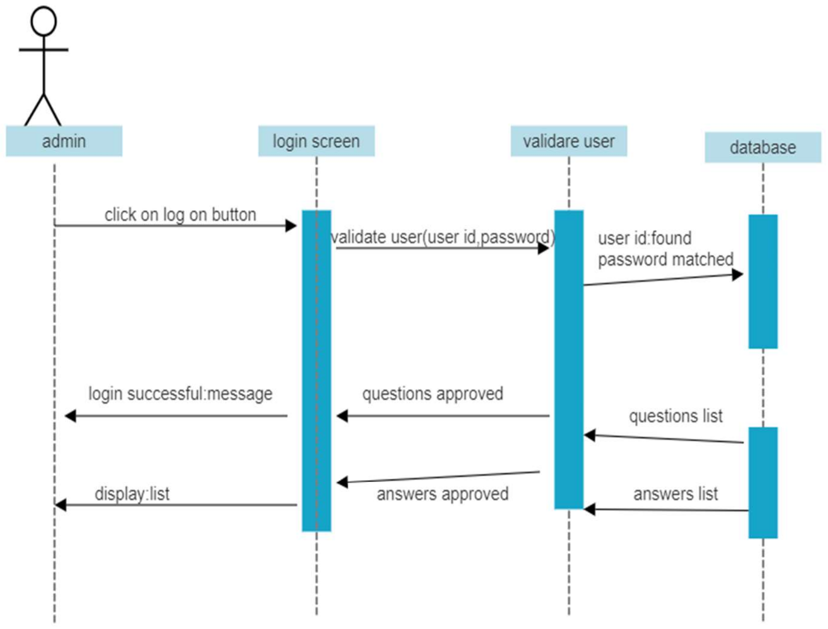


Fig No.9 Admin Sequence Diagram

ER DIAGRAM

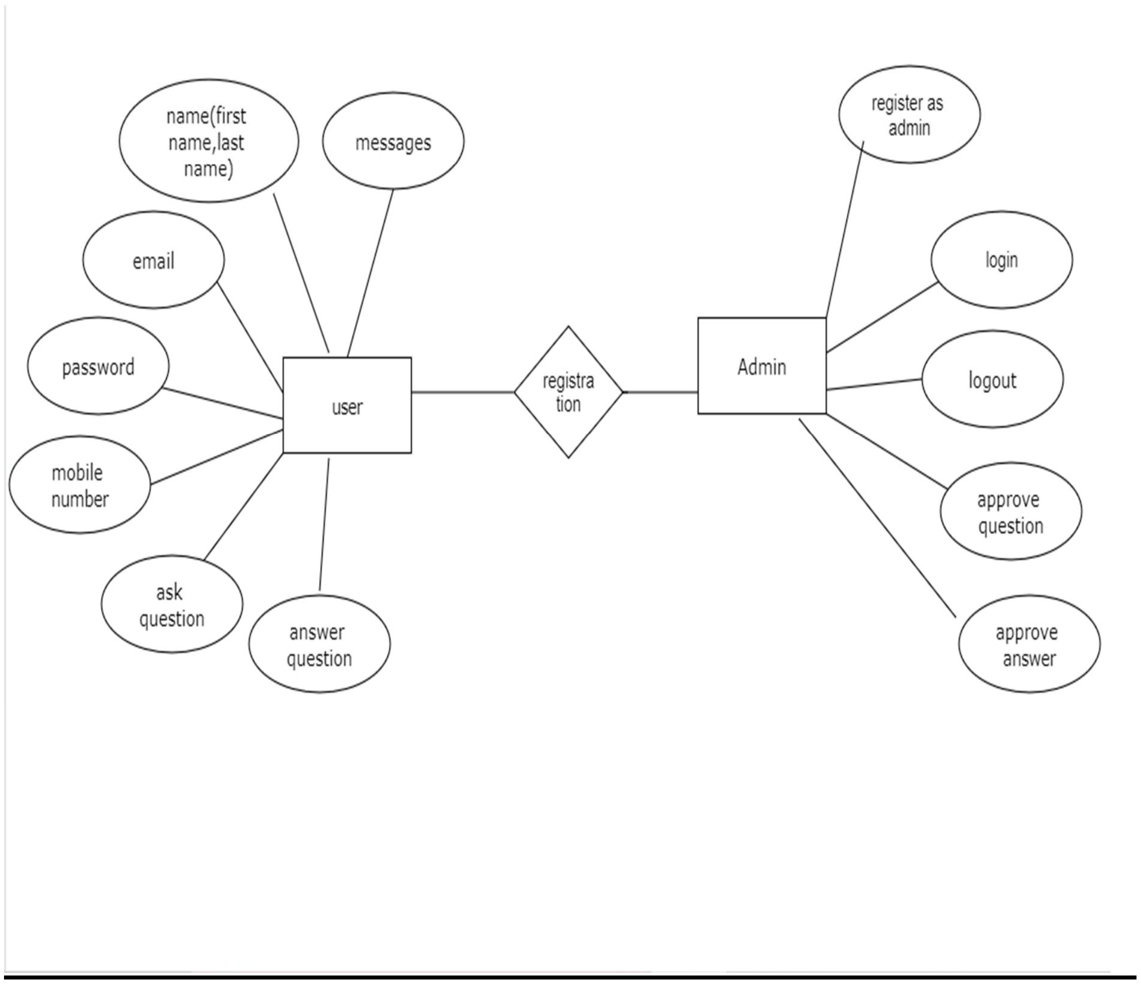
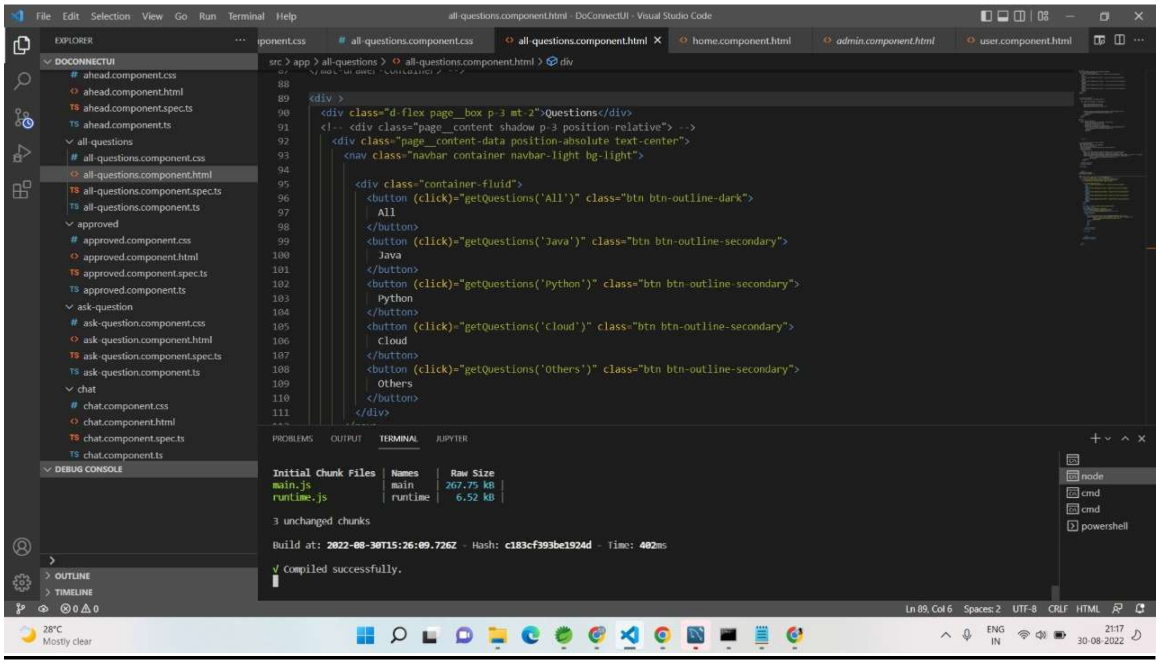


Fig. No.10 Er Diagram

PROJECT SCREENSHOTS

1. FRONTEND CODE

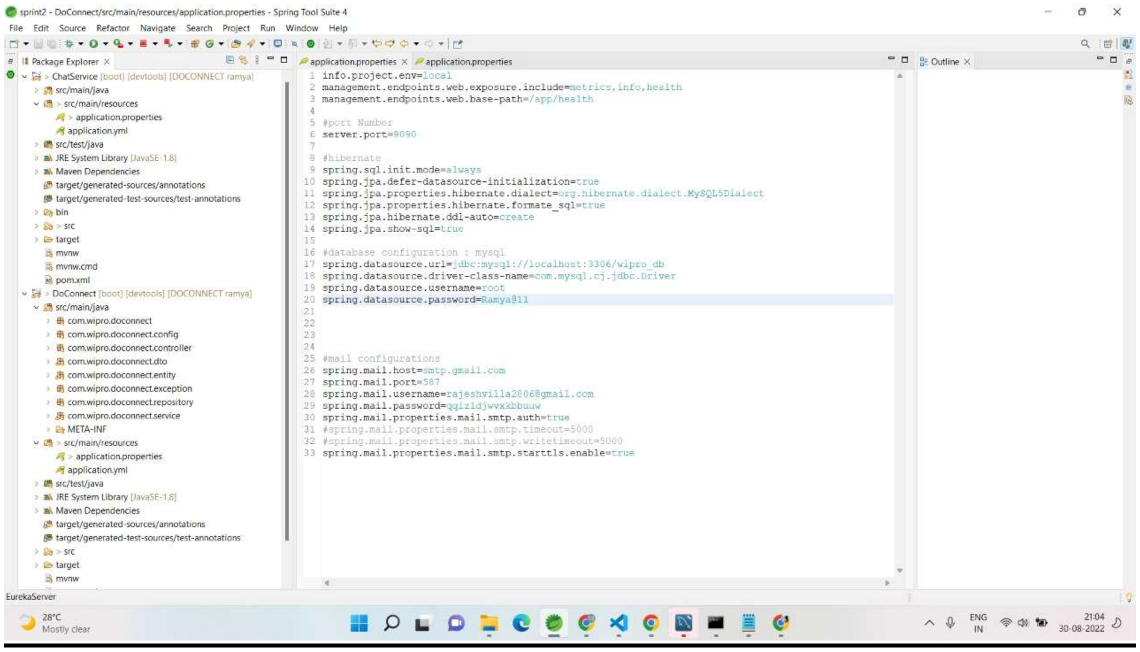


```html
<div >
  <div class="d-flex page__box p-3 mt-2">Questions</div>
  <!-- <div class="page__content shadow p-3 position-relative"> -->
    <div class="page__content-data position-absolute text-center">
      <nav class="navbar container navbar-light bg-light">

        <div class="container-fluid">
          <button (click)="getQuestions('All')" class="btn btn-outline-dark">
            All
          </button>
          <button (click)="getQuestions('Java')" class="btn btn-outline-secondary">
            Java
          </button>
          <button (click)="getQuestions('Python')" class="btn btn-outline-secondary">
            Python
          </button>
          <button (click)="getQuestions('Cloud')" class="btn btn-outline-secondary">
            Cloud
          </button>
          <button (click)="getQuestions('Others')" class="btn btn-outline-secondary">
            Others
          </button>
        </div>
```

```
Initial Chunk Files | Names   |  Raw Size
main.js             | main    | 267.75 kB |
runtime.js          | runtime |   6.52 kB |

3 unchanged chunks

Build at: 2022-08-30T15:26:09.726Z - Hash: c183cf393be1924d - Time: 402ms

√ Compiled successfully.
```

Fig no.11 Frontend code

2. BACKEND CODE



```
info.project.env=local
management.endpoints.web.exposure.include=metrics,info,health
management.endpoints.web.base-path=/app/health

#port Number
server.port=9090

#hibernate
spring.sql.init.mode=always
spring.jpa.defer-datasource-initialization=true
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect
spring.jpa.properties.hibernate.formate_sql=true
spring.jpa.hibernate.ddl-auto=create
spring.jpa.show-sql=true

#database configuration : mysql
spring.datasource.url=jdbc:mysql://localhost:3306/wipro_db
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.username=root
spring.datasource.password=Ramya@11




#mail configurations
spring.mail.host=smtp.gmail.com
spring.mail.port=587
spring.mail.username=rajeshvilla2806@gmail.com
spring.mail.password=qqlzldjwvxkbbuuw
spring.mail.properties.mail.smtp.auth=true
#spring.mail.properties.mail.smtp.timeout=5000
#spring.mail.properties.mail.smtp.writetimeout=5000
spring.mail.properties.mail.smtp.starttls.enable=true
```

Fig no.12 Backend code

3. ADMIN REGISTER PAGE WITH TOKEN

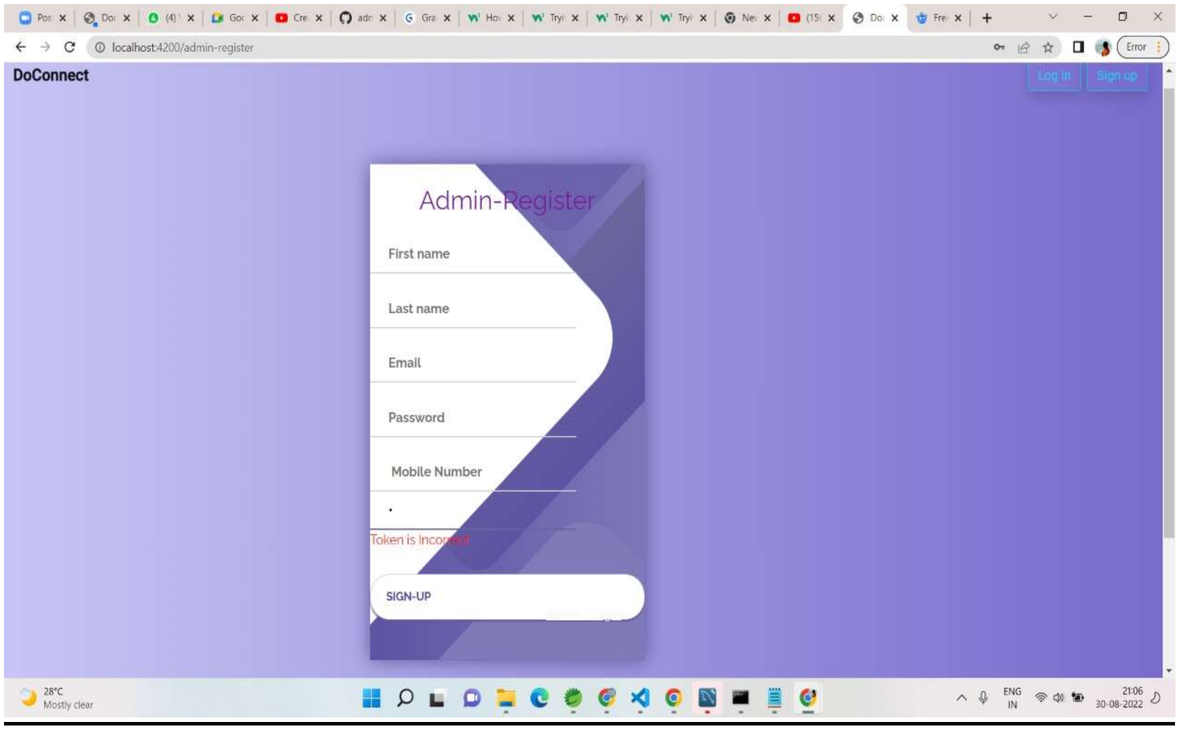


Fig no.13 Admin register page with token

4. USER LOGIN

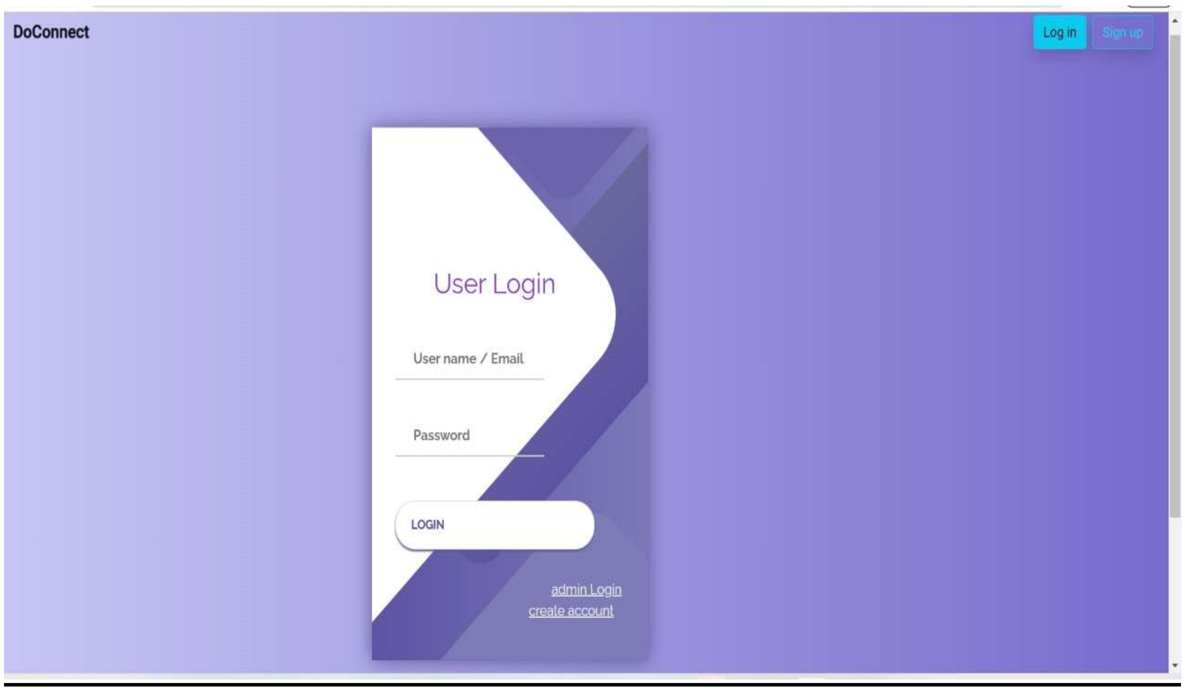
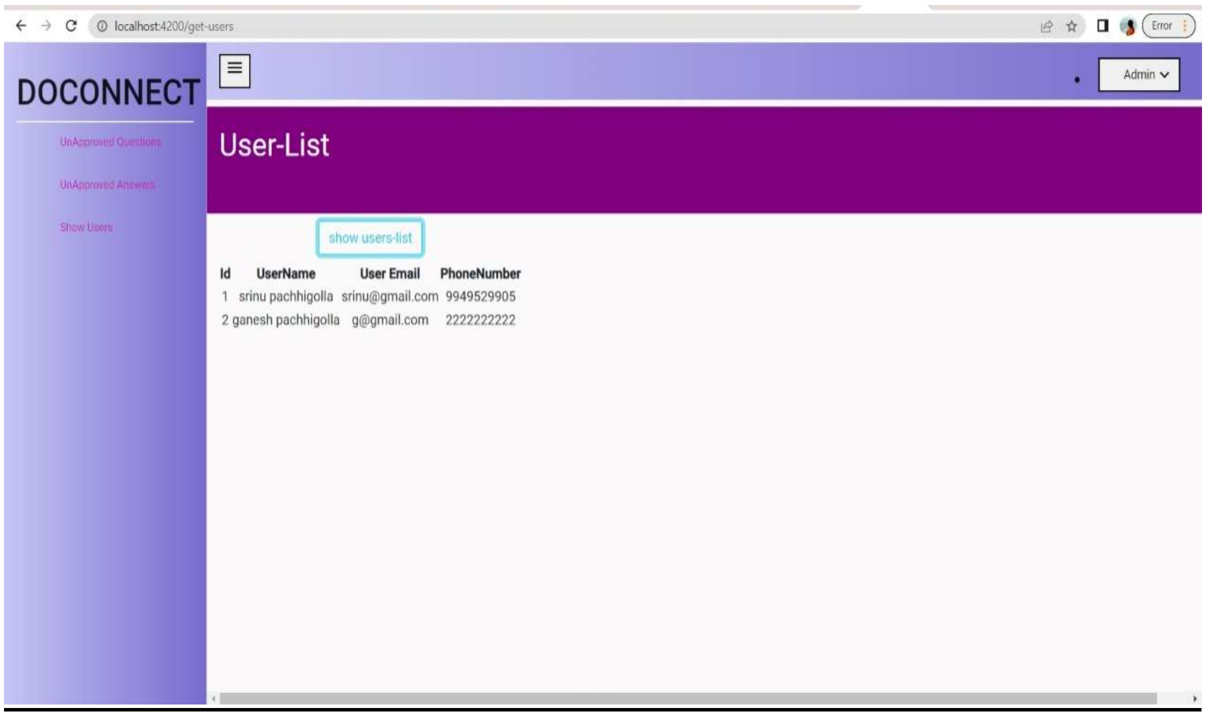


Fig no.14 User login

5. ADMIN DASHBOARD



Id	UserName	User Email	PhoneNumber
1	srinu pachhigolla	srinu@gmail.com	9949529905
2	ganesh pachhigolla	g@gmail.com	2222222222

Fig no.15 Admin dashboard

6. USER DASHBOARD

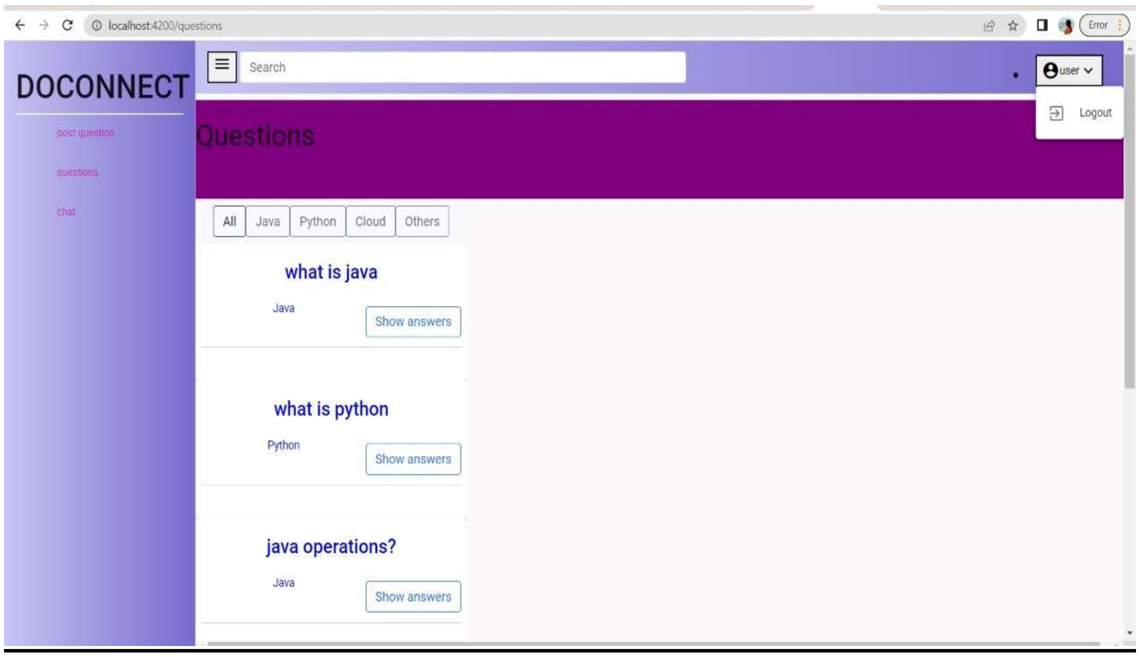


Fig no.16 User dashboard