**CAPSTONE PROJECT**

**DOCONNECT**

**PROJECT TEAM**

**VISHAL BHARTI (TEAM LEADER)**

**ALOK SINHA**

**NEERAJ SHEKHAWAT**

**SPOORTHI**

Contents

[1 Introduction 3](#_Toc92460813)

[2 Purpose 3](#_Toc92460814)

[3 System Features 5](#_Toc92460815)

[3.1 Homepage 6](#_Toc92460816)

[3.2 User Interface 7](#_Toc92460817)

[3.3 Admin Interface 11](#_Toc92460818)

[4 Database Schema 16](#_Toc92460819)

[5 ER Diagram And Flow Diagrams 18](#_Toc92460820)

# Introduction

This document is prepared in order to determine a software requirement specification for DoConnect. DoConnect is a social network question and answer system on which people ask various question regarding their queries and the people across the globe can answer the question. Also it is a kind of community which improves the knowledge of the user in every field. In order to gain an overview about the report, the purpose of this document will be given, then an overall description of DoConnect system is followed. In addition to these, system features such as post question, response, login, logout etc functionality that are described . After mentioning about the introduction of the specific requirements , database schema and E-R Diagram also will be addressed for it.

# Purpose

The SRS is needed to evolve as the development of the software product processes. The purpose of this document is to give a complete description about how DoConnect social network system can be developed. This document is to provide information about what the software product is to do to customers and establish a communication between customers and also become helpful for development. In addition to these, it provide a basis for validation and verification. The issues which are basically addressed are functionality such as posting questions, response, login ,sign up, search, external interfaces, performance, attributes and the design constraints of the system.

**Specific Requirements**

**Functional Requirements**

The requirements for the software system are as follows

**Functional Requirement 1**

* Description: Login
* Dropdown: User and Admin login
* Input: Email and Password are Entered
* Processing : Verifying the email from database
* Output : Login Successful

**Functional Requirement 2**

* Description: Sign up
* Dropdown: User and Admin sign up
* Input: Name, Email and password are entered
* Output: Register Successful

**Functional Requirement 3**

* Description: Post Question
* Input : Question can be post with topic after login
* Processing: Admin will verify the question ,then it will be displayed
* Output: Question will be displayed

**Functional Requirement 4**

* Description: Response
* Input : Can answer the question in text editor
* Processing: Admin will first approve the response
* Output: Response will be displayed

**Functional Requirement 5**

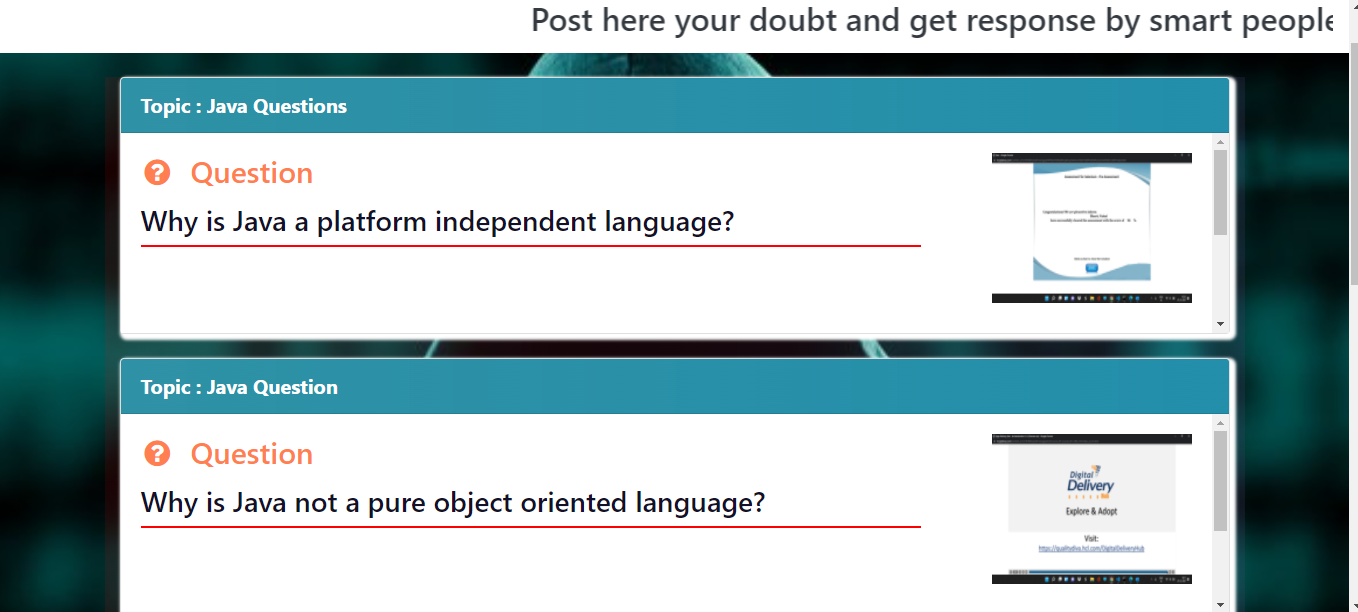
* Description: Search
* Input : Search the specific topic or question
* Processing : Search according to topic and question
* Output : Result will be shown

# System Features

In this section, all normal and alternative flow of events are organized with the assumption that users or administrators are successfully reached their homepage by logging into the system. This assumption is made in order to describe specifications of the sub-features.

## Homepage

Homepage is the first landing page of DoConnect. It contains different functionality like login, sign up, post question, search box. We can login as user and admin both. For registration we can also sign up as both user and admin. Homepage body contains questions of different topics. We can response to the questions after login first as user. We can login as user and admin both. For registration we can also sign up as both user and admin. To post questions we can go to post question icon but before that we have to login first. We can also search according to any topic and question in the search box.

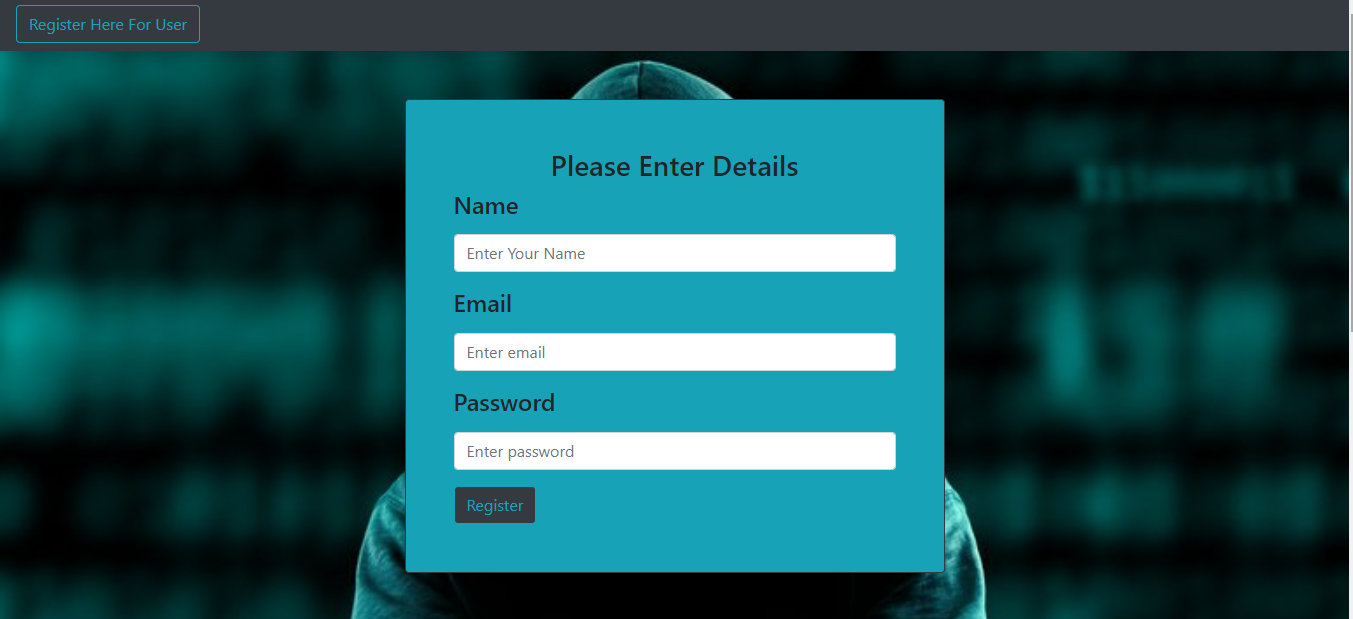
* Homepage landing page is shown as below.It is the first landing page of DoConnect .It contains different functionality like login, sign up, post question, search box. The body contains all the questions with their topics and answers.

## User Interface

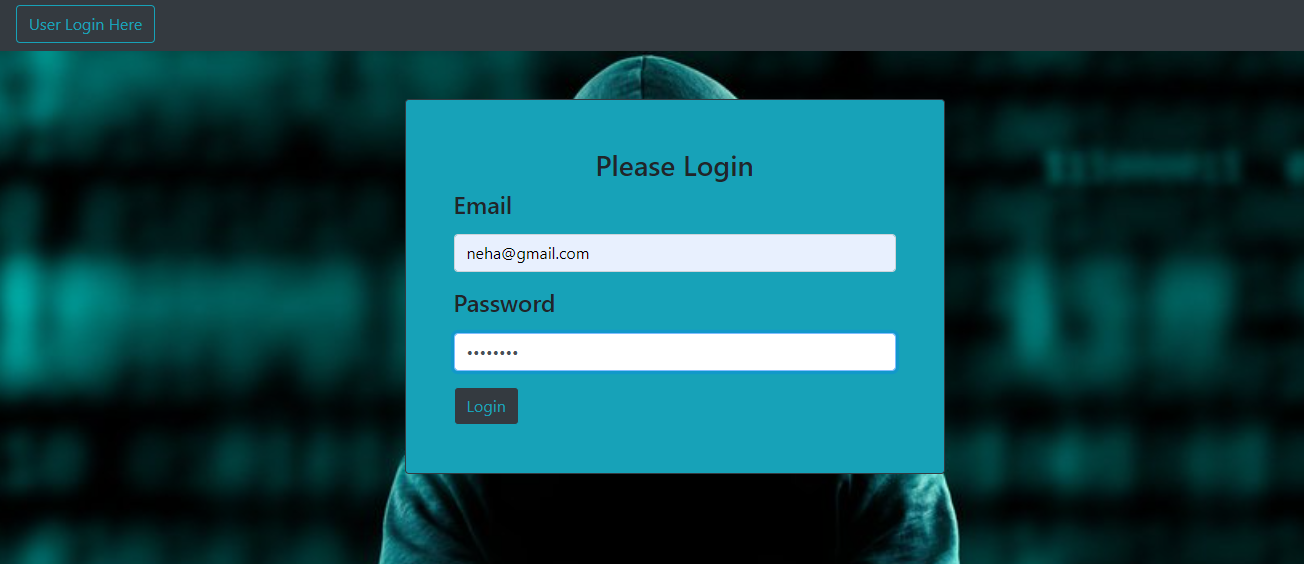
User Interface contains my post functionality, by going in it user can see the

Questions posted by them. The body contains different questions with their topic. To answer any question user can go to response button and can answer them. User can answer any question multiple times and user can also give response to any answer. To post question user can go to post question icon and post any question there with its topic and user can see the question after its gets admin approval. User can also chat with any user one to one . User can also logout by going to logout button.

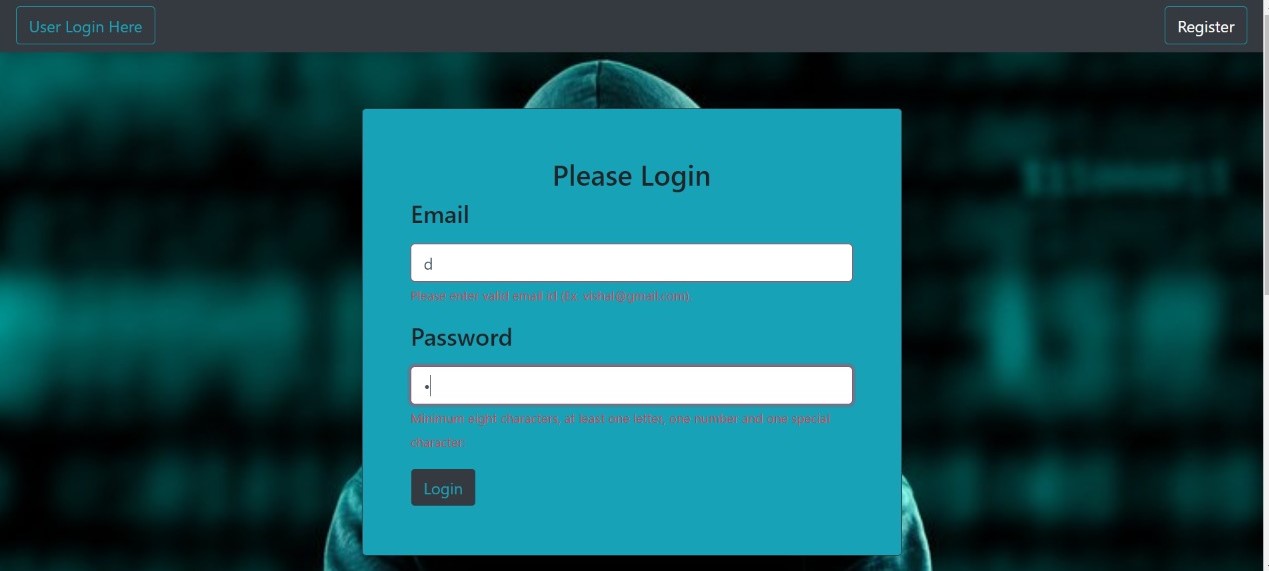
* User can go to user sign up in the sign up section of the homepage to register themselves which will redirects them to user sign up page where user have to enter name , email , password to register themselves.



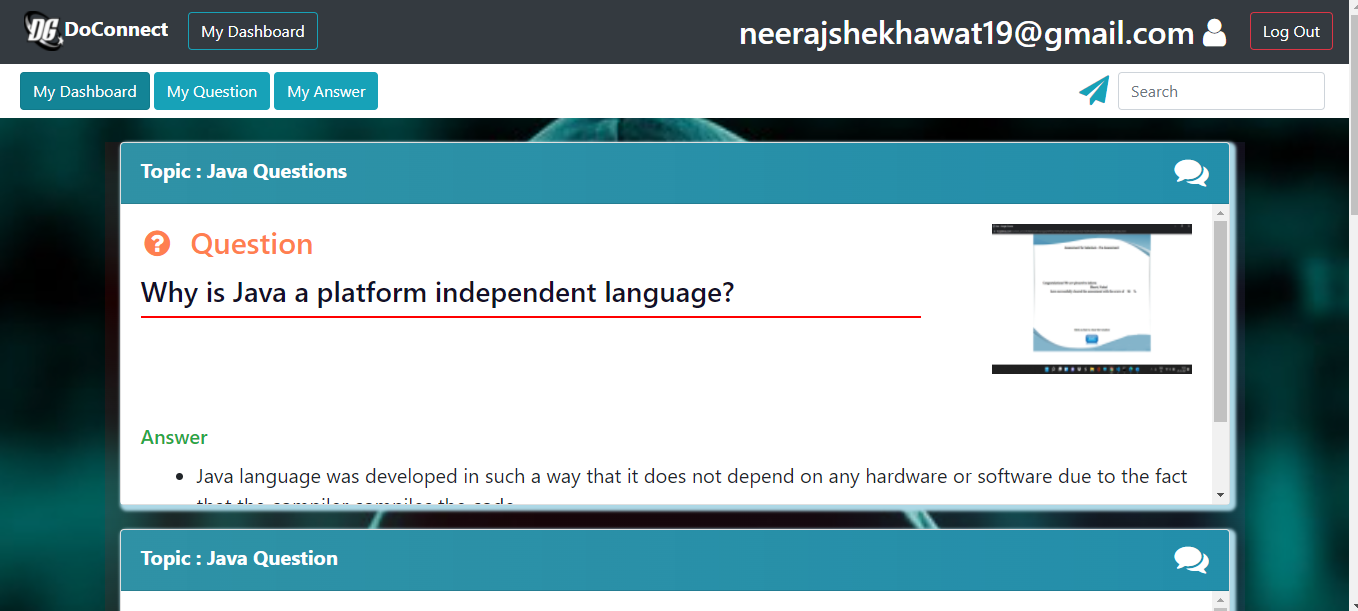
* When we go to user login in login section of dashboard, it directs to the login section for user where user can enter their email and password to login.

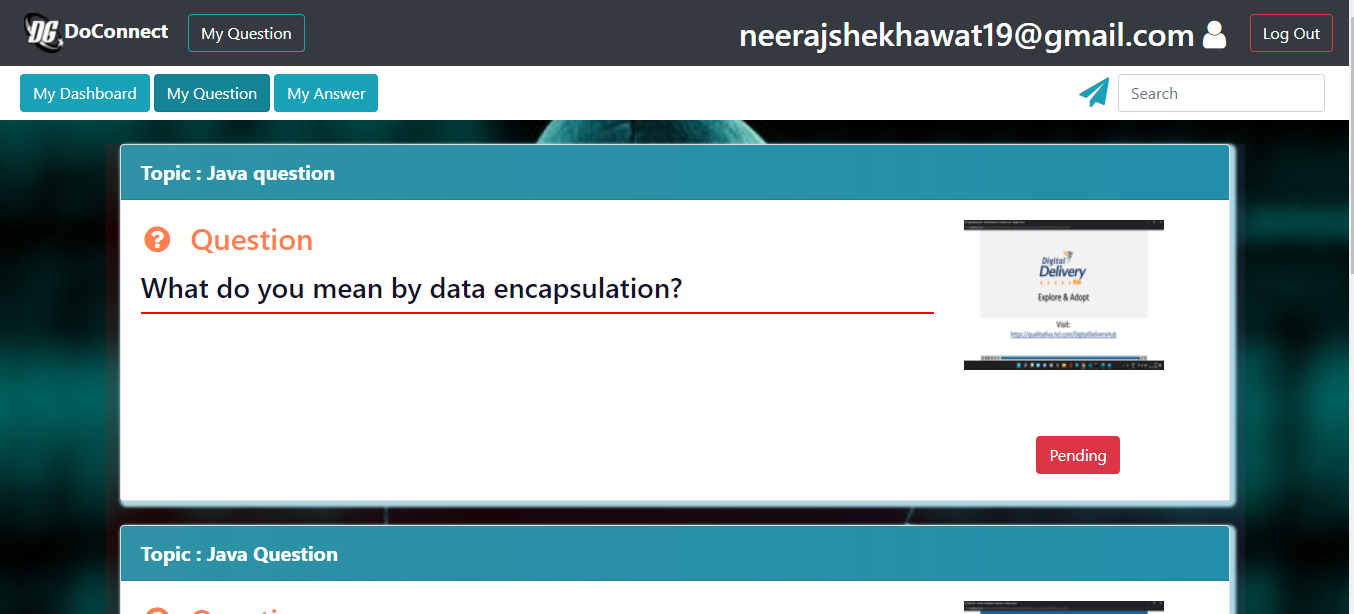
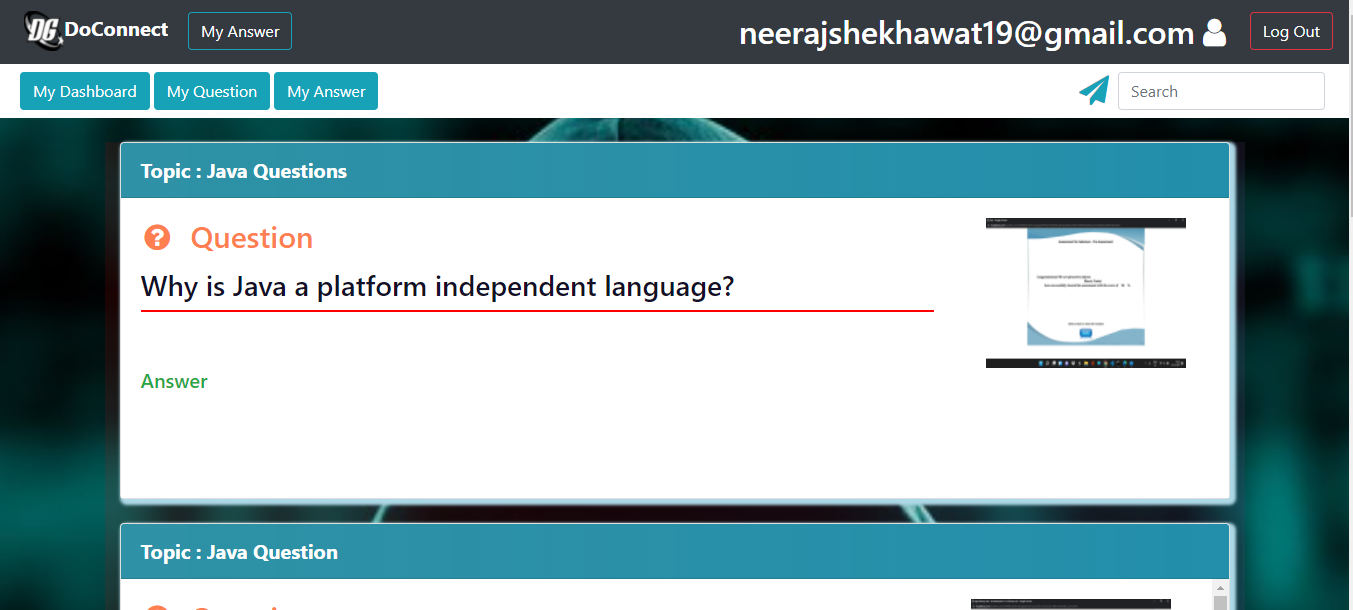
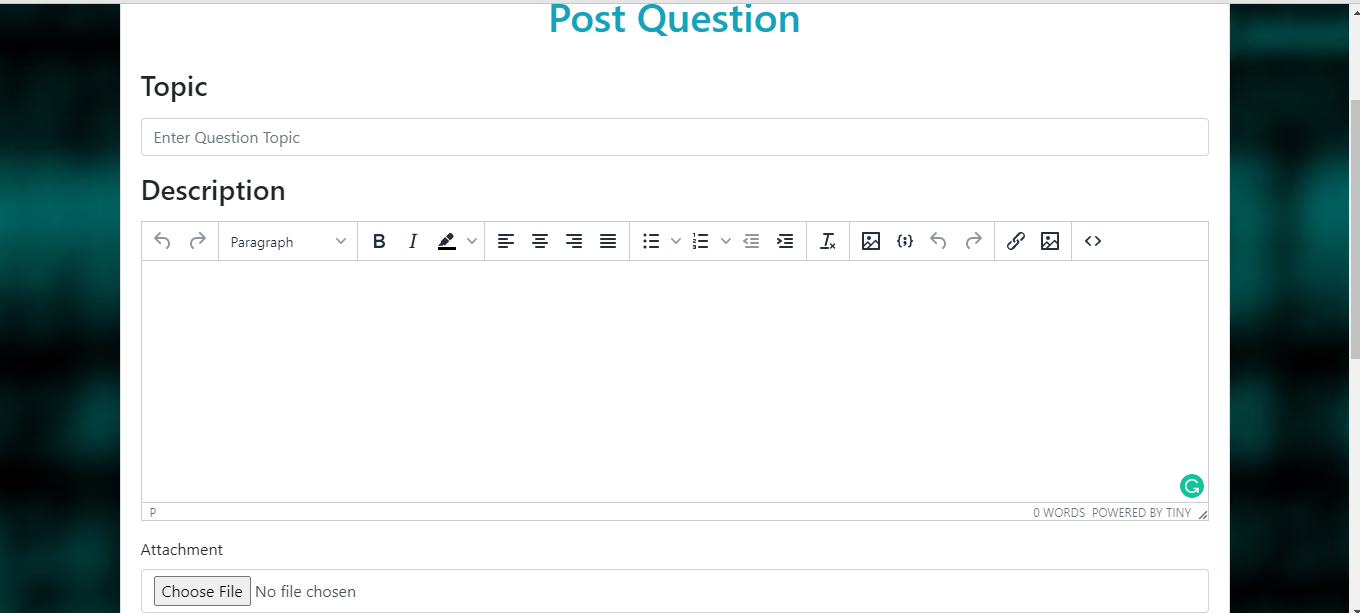


* User have to follow the validations points to login in. If user enter wrong credentials it will show error as shown below



* User Dasboard landing page is shown as below. User Dashboard body contains all the questions with their answers and contains different functionality like post questions, response and my post.

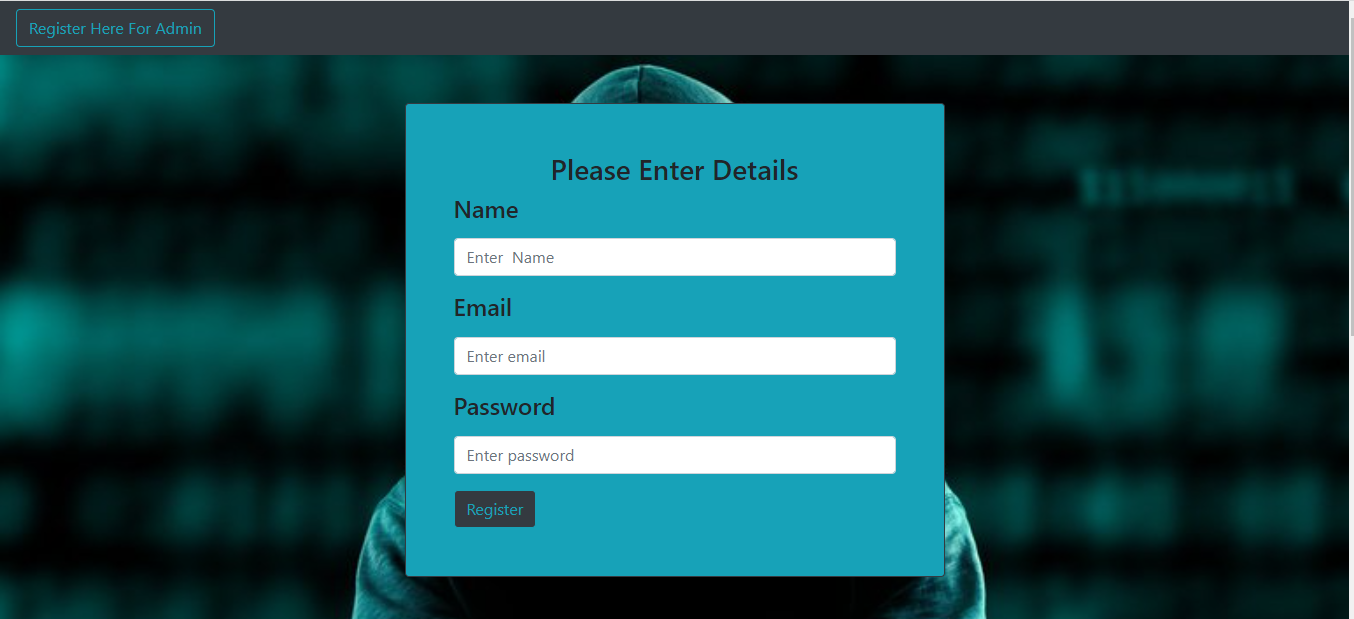


* User can to My Question to see all the questions posted by user and he can check the status of the questions whether they are pending or not.
* User can go to My Answer to see all the answers posted by them and they can check the status of the answers whether they are pending or not.
* To post any question user can go to post question where user can write topic of the question and can write the question in the text editor .

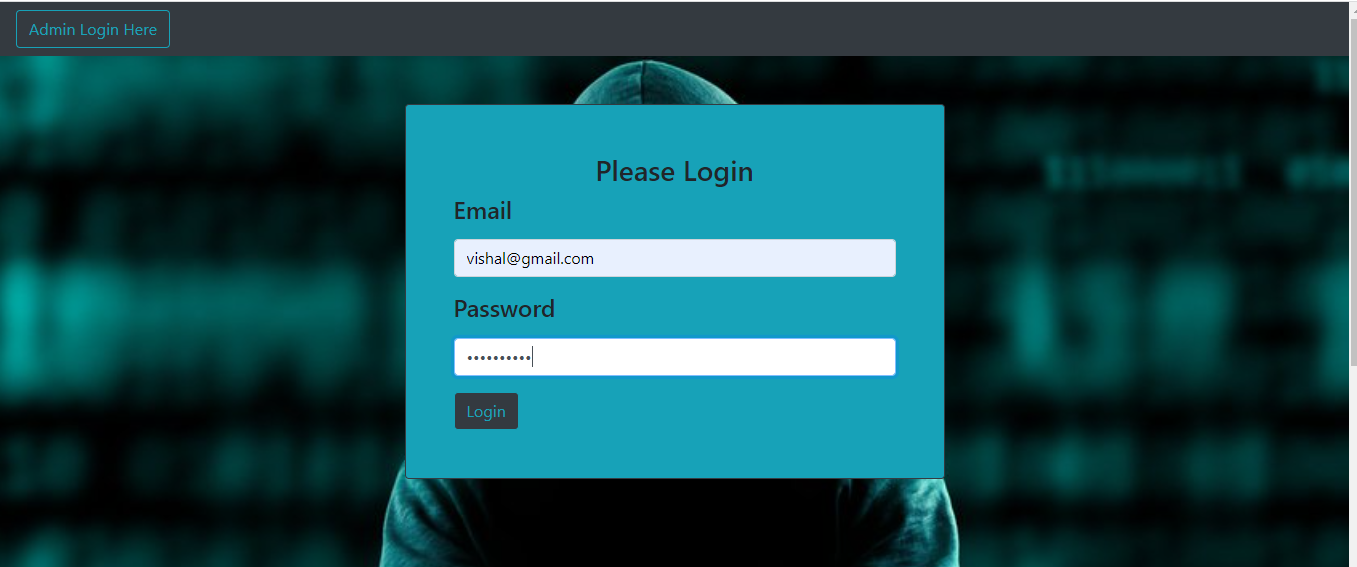
## Admin Interface

Admin interface contains different functionality like Admin dashboard, post question, search box, pending requests, users. The body contains all the questions of different topics. Admin can approve or discard any response made by the user if its violates the validation points. Admin can also delete any user and can perform edit operation on users by going to user button. Admin can also add any user by going to create user. Whenever any user post question r give any response admin will receive a mail for it and after that going to the pending post on his dashboard.

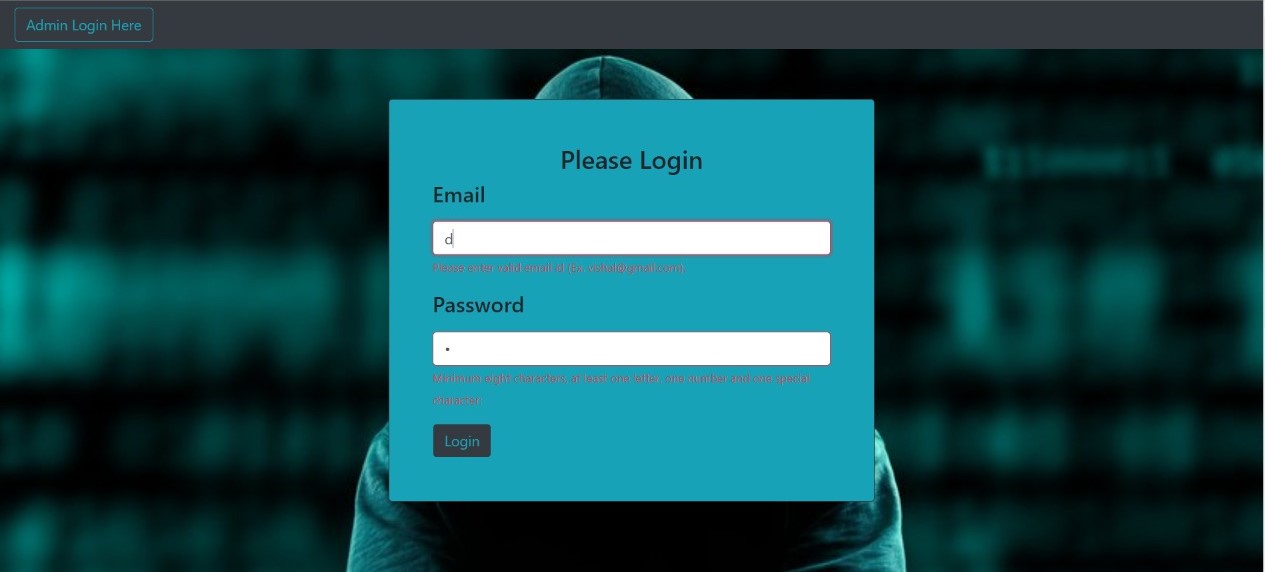
* Admin can go to admin sign up in the sign up section of the homepage to register themselves which will redirects them to user sign up page where user have to enter name , email , password to register themselves.



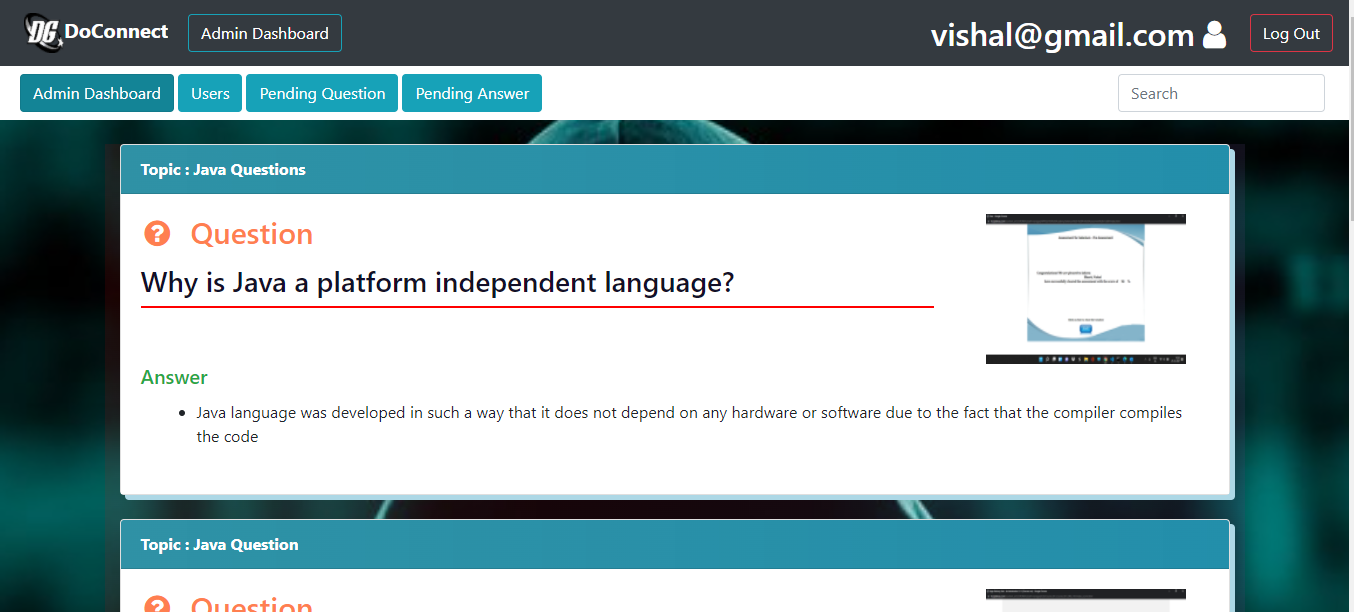
* To login as admin, admin has to go to Admin login in the login section of the homepage which will directs admin to admin login page where admin can enter their email and password.



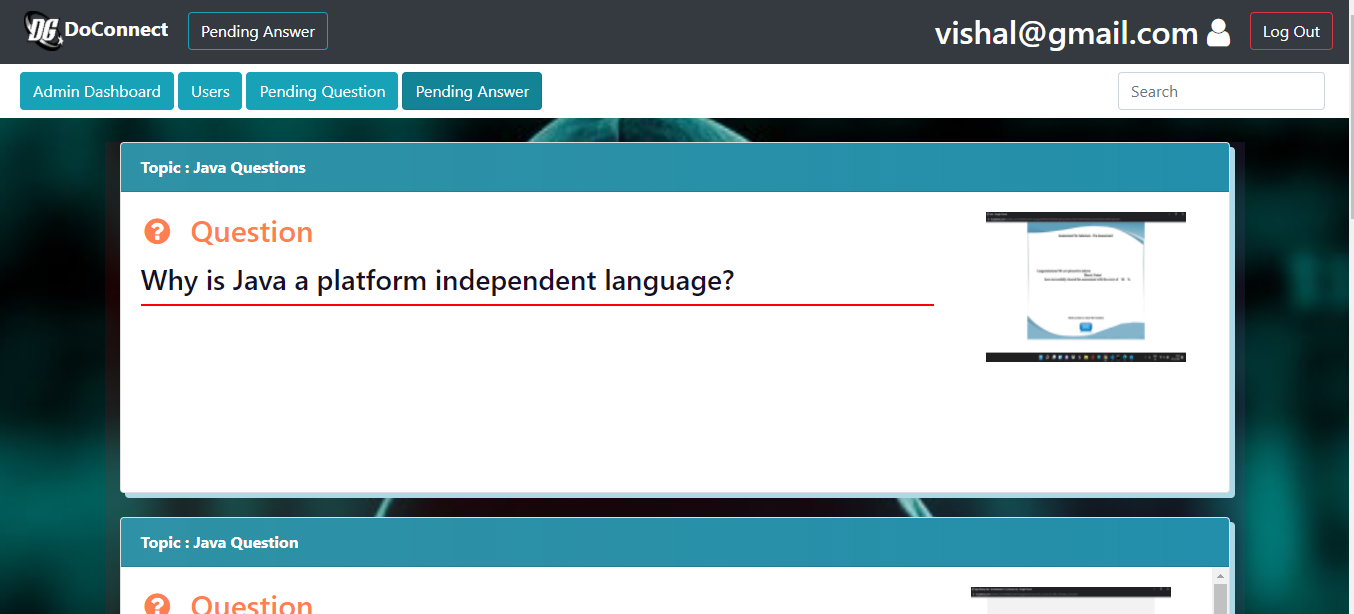
* User have to follow the validations points to login in. If user enter wrong credentials it will show error as shown below



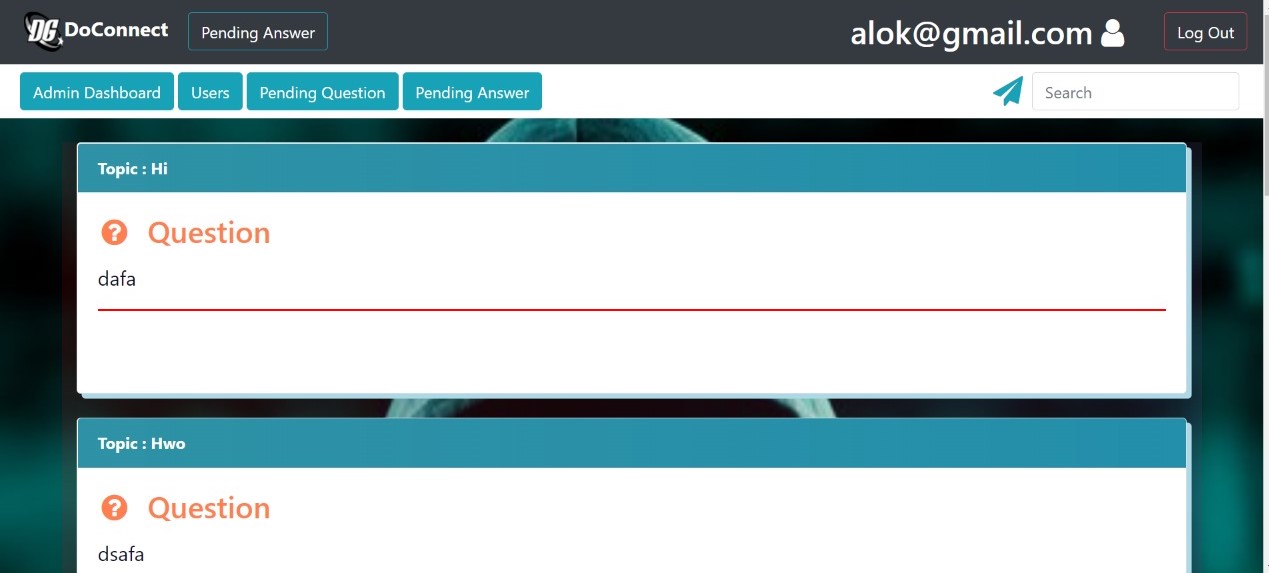
* After login admin dashboard will gets opened where admin can see all the questions with their answers and admin dashboard contains different functionality like post question, pending post, users etc.

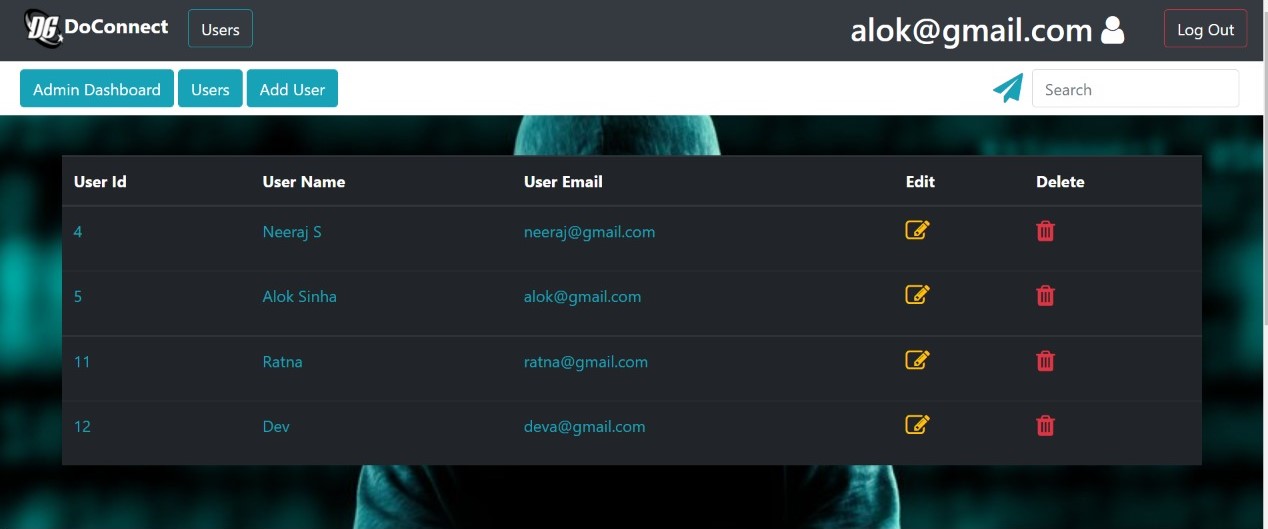


* When admin go to pending question, they can see all the pending questions and answers posted by user which admin can approve or discard according to its validation. After approving to the post, it will add to the dashboard.



* When admin go to pending answer, they can see all the pending answers posted by user which admin can approve or discard according to its validation. After approving to the post, it will add to the dashboard.



* Admin can go to the users section to add, delete and to perform the edit operations on users.

# Database Schema

1. **User bean**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID (Primary Key) | ADMIN | Email | Name | password | username |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

1. **Question Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Que\_id(Primary key) | Approval | Description | Img\_url | topic | User\_id |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

1. **Question\_Answer Table**

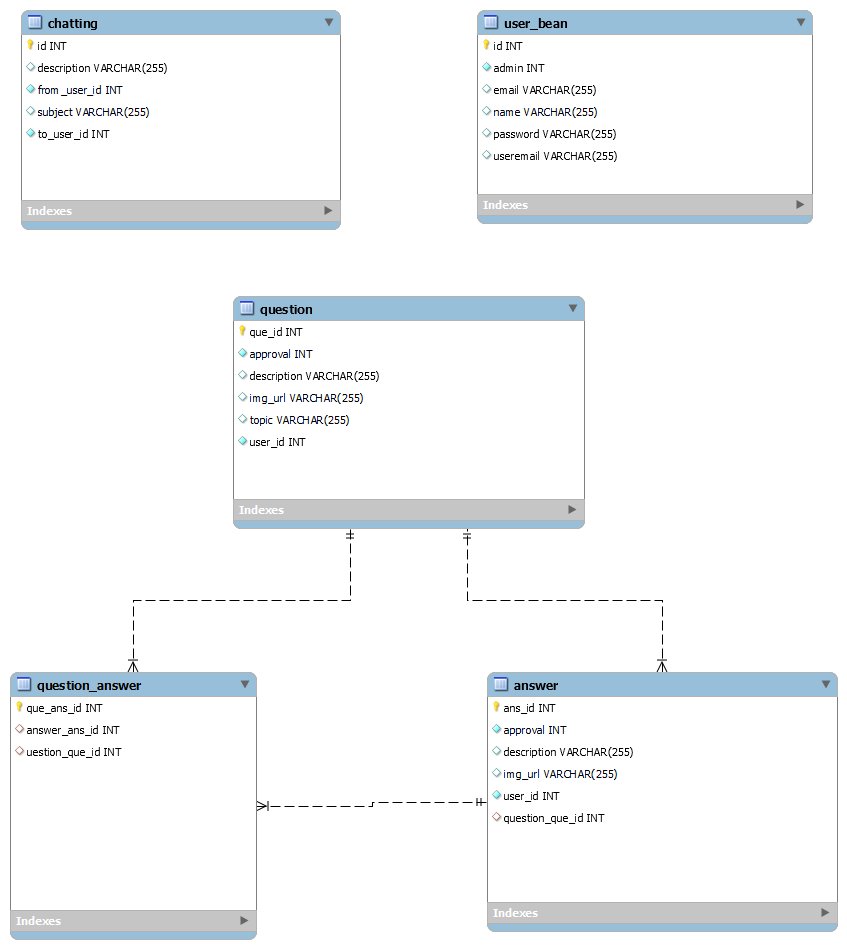
|  |  |  |
| --- | --- | --- |
| Que\_ans\_id(Primary key) | Answer\_ans\_id | Question\_que\_id |
|  |  |  |
|  |  |  |

1. **Answer Table**

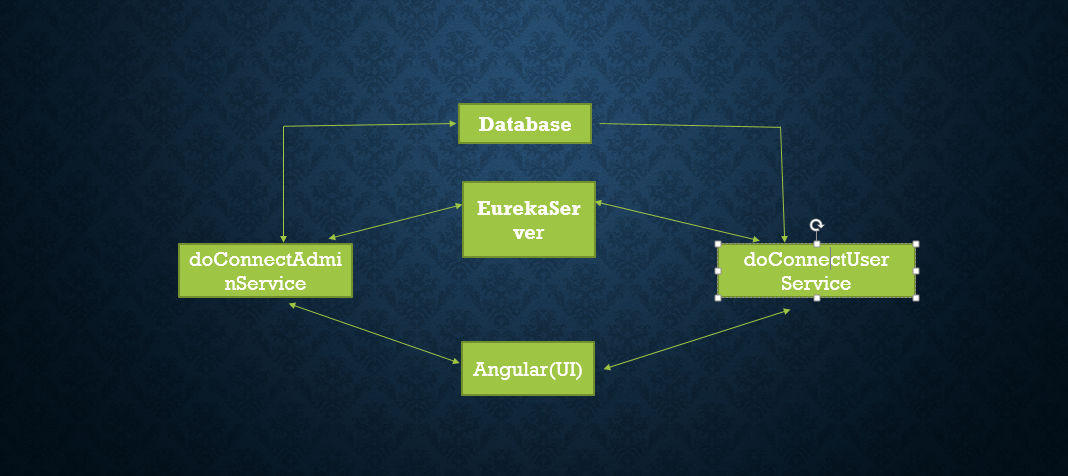
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ans\_id(Primary key) | Approval | description | Img\_url | User\_id | Question\_que\_id |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

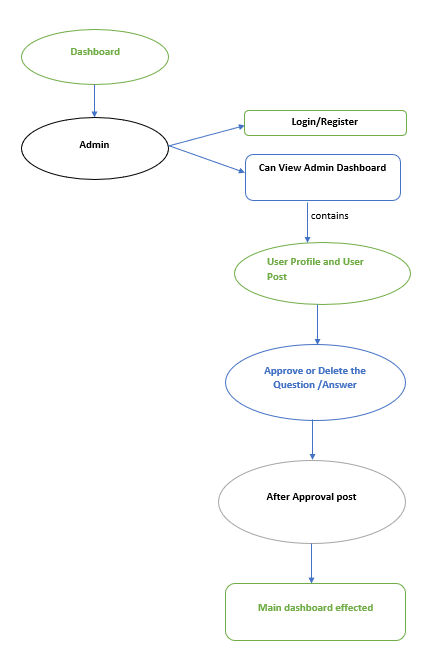
# ER Diagram And Flow Diagrams

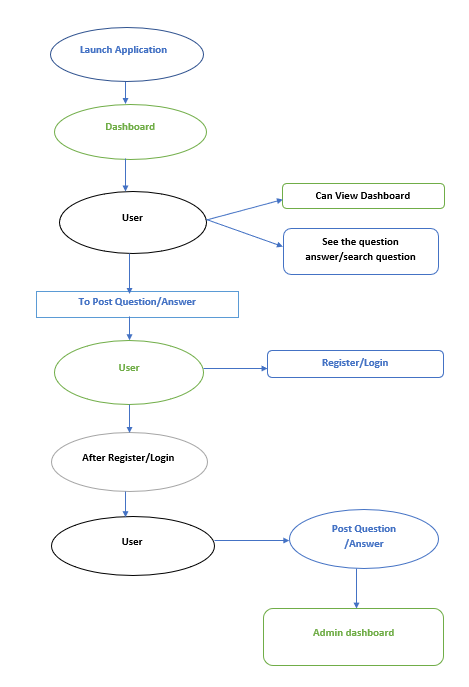
E-R Diagram



Microservices Structure



 Admin Flow Diagram

User Flow Diagram