*A Project Report on*

IOPE Diaries

*Submitted by:*

**PAWAR SEJAL RAVINDRA** (2030408246063)

**JADHAV MANISH SHASHIKANT** (2030408246006)

**GURAV SHRUTI SANJAY** (2030408246053)

*Under the guidance of*

### Prof. A. H. Shelake

*in fulfilment for the award of*

## DIPLOMA IN

**INFORMATION TECHNOLOGY**

2022-23

at



### Dr. Babasaheb Ambedkar Technological University

### (Diploma Wing)

Lonere, Tal. Mangaon, Dist. Raigad, Maharashtra (INDIA) – 402103

This is to certify that, the Project entitled *‘IOPE Diaries’* is a work carried out by

**PAWAR SEJAL RAVINDRA** (2030408246063)

**JADHAV MANISH SHASHIKANT** (2030408246006)

**GURAV SHRUTI SANJAY** (2030408246053)

the students of Third Year of Diploma in Information Technology at Dr. Babasaheb Ambedkar Technological University (Diploma Wing),Lonere, during the academic year 2022 - 2023 in the fulfillment of the requirements for the award of Diploma in Information Technology.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Prof. A. H. Shelake Prof. K. R. Korpe Prof. S. M. Gaikwad**

**(Project Guide) (Project Coordinator) (Head of Department[I.T.])**

**Place:** Lonere

**Date:**

The spread of pandemic COVID-19 has drastically disrupted every aspect of human life including education. It has created an unprecedented test on education. In many educational institutions around the world, campuses are closed and teaching-learning has moved online. Internationalization has slowed down considerably. In India, about 32 crore learners stopped to move schools/colleges and all educational activities ended. The closure of colleges and universities has widespread individual, organizational, and learning and teaching implications for students, faculty, administrators, and the institutions themselves.

Lack of access to technology or fast, reliable internet access can prevent students in rural areas and from disadvantaged families. Lack of access to technology or good internet connectivity is an obstacle to continued learning. Teachers have reported the Institute closures negatively impact student learning outcomes, that students are more likely to complete assignments if they have access to internet at home. Since lack of digital mediums among the student, despite of their geographical locations. It became much more difficult for them to acquire practical knowledge in their respective fields. Online learning only drove the students through theoretical knowledge, losing one of the most important skills like problem solving, practical programming skills, etc.

The ultimate goal of out project is to address all the problems faced by students as well as faculties too. Our project focuses developing a website which can run on almost any devices, where students with underprivileged backgrounds will gain interest in programming, learn and develop as well as practice their problem-solving skills and of course increase stats to build skills for real world challenges.

Firstly, we are indebtedly grateful to our Head of Department **Prof. S. M. Gaikwad** for his support. We are also grateful to our guide **Prof. A. H. Shelake** and our Project Coordinator **Prof. K. R. Korpe** for their valuable guidance and support. Without their support this project would not have been initiated.

We deeply obliged to thank all the faculty members of **Department of Information Technology**, for their valuable guidance and constant monitoring throughout the work of our project. Also, we would like to thank our college for providing us all the necessary resources for the project.

We, especially we are very thankful to our team members, for their contribution to the project work, kind co-operation and encouragement which help to developing the project.

Thank You!

**PAWAR SEJAL RAVINDRA** (2030408246063)

**JADHAV MANISH SHASHIKANT** (2030408246006)

**GURAV SHRUTI SANJAY** (2030408246053)

Department of Information Technology,

Dr. Babasaheb Ambedkar Technological University (Diploma Wing),

Lonere - Raigad

CERTIFICATE ii

ABSTRACT iii

ACKNOWLEDGEMENT iv

1. [INTRODUCTION 7](#_bookmark0)
   1. [Problem Definition](#_bookmark1) 7
   2. [Scope & Objective](#_bookmark3) 7
   3. [Language & Tools](#_bookmark2) 8
   4. [System Requirements](#_bookmark2) 8
2. [LITERATURE SURVEY](#_bookmark4) 9
   1. Existing Systems [9](#_bookmark5)
   2. [Flaws in Existing Systems](#_bookmark2) 10
   3. [Proposed System](#_bookmark2) 10
3. [REQUIREMENT ANALYSIS 11](#_bookmark6)
   1. [Functional Requirements 11](#_bookmark7)
   2. [Non-Functional Requirements 11](#_bookmark8)
   3. [Test Criteria 12](#_bookmark9)
4. [DESIGN 1](#_bookmark10)3
   1. [Software Development Life-Cycle (SDLC) Model](#_bookmark11) 13
   2. [Flow Charts](#_bookmark2) 14
5. [User Interface (UI) 19](#_TOC_250004)
6. [Source Code Snippets 23](#_TOC_250003)
7. [Meet the TEAM 27](#_TOC_250002)

[CONCLUSION 29](#_TOC_250001)

[REFERENCES 30](#_TOC_250000)

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.** | **Fig. No.** | **Name of the Figure** | **Page No.** |
| 1. | 4.1 | Iterative Model (SDLC) | 14 |
| 2. | 4.2 | Data Flow Diagram | 15 |
| 3. | 4.3 | Class Diagram | 15 |
| 4. | 4.4 | Entity-Relationship Diagram | 16 |
| 5. | 4.5 | Control Flow Diagram | 17 |
| 6. | 4.6 | Use Case Diagram | 18 |
| 7. | 4.7 | System Architecture | 19 |
| 8. | 5.1 | Sign-Up Page | 20 |
| 9. | 5.2 | Login Page | 20 |
| 10. | 5.3 | Home Page | 21 |
| 11. | 5.4 | Question Page | 21 |
| 12. | 5.5.1 | Coding Page | 22 |
| 13. | 5.5.2 | Coding Page | 22 |
| 14. | 5.6 | About-Us Page | 23 |
| 15. | 6.1 | Python Server | 24 |
| 16. | 6.2 | pymongo | 24 |
| 17. | 6.3 | Postman API | 25 |
| 18. | 6.4.1 | Database | 26 |
| 19. | 6.4.2 | Database | 26 |
| 20. | 6.5.1 | Sign-up Logic | 27 |
| 21. | 6.5.2 | Sign-up Logic | 27 |
| 22. | 7.1 | Project Meetings | 29 |

## LIST OF TABLES

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.** | **Tab. No.** | **Name of the Table** | **Page No.** |
| 1. | 3.1 | Test Criteria Table | 12 |

## INTRODUCTION

#### “IOPE Diaries”



**IOPE Diaries**

#### Problem Definition:

* + - **Problem Statement:** Designing an effective college event website that caters to the needs of both students and faculty, while ensuring seamless navigation and optimal user experience, is a challenging task. The website must provide comprehensive information about conducted events, relevant details, and should also incorporate features that promote engagement, networking, and community building. The challenge lies in balancing functionality, aesthetics, and usability, while adhering to time and budget constraints.
    - **Relevance:** A college event website is highly relevant as it serves as a centralized hub of information for students, faculty, and staff regarding conducted events and activities on campus. It allows individuals to easily access important details such as event dates, times, locations, and registration information. Additionally, it provides a platform for students to engage with each other and the campus community, fostering a sense of belonging and promoting a vibrant campus culture. By promoting and highlighting the diverse range of events and activities on campus, the website encourages participation, collaboration, and lifelong learning.

#### Scope & Objective:

* + - **Scope:**

1. **Accessibility:** As it is a web-based platform, it can be accessed anytime.
2. **Free-of-cost:** Does not charge for anything.
3. **Event Promotion:** One key scope of a college event website is to promote upcoming events on campus.
4. **Analytics and Reporting:** Analytics and reporting can also help demonstrate the impact of events on the campus community and the broader public.

#### C:\Users\tejas\Downloads\start-up.pngObjectives:

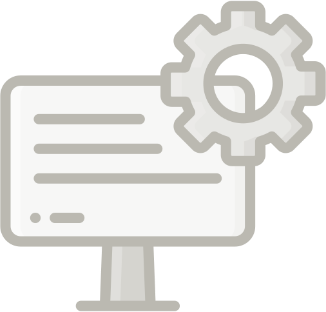
1. **Easy-to-use:** Ease of accessing the platform.
2. **Usability:** User Friendly Interface.
3. **Gain:** Confidence to participate in other events.
4. **Promote Campus Culture:** A third objective for a college event website could be to promote campus culture and showcase the diversity of events and activities happening on campus. By highlighting events from a variety of departments and student groups, the website can help create a more vibrant and inclusive campus community.

#### Languages & Tools:

1. **Front End:** HTML, CSS, JavaScript.
2. **Back End:** PHP, MySQL.
3. **Operating System:** Windows 11.
4. **Web Browser:** Google Chrome.
5. **IDE used:** Visual Studio Code 1.63.0
6. **Tools:** Bootstrap.



#### System Requirements:

1. **Screen Resolution:** 1280x1024 or larger
2. **Application Window:** 1024x680 or larger
3. **Operating System:** Windows 8 or later, macOS Sierra 10.12 or later, Ubuntu 14.04+
4. **Web Browser:** Google Chrome (Recommended)
5. **Internet Connection:** Required
6. **Memory:** 2GB Minimum, 4GB Recommended

## LITERATURE SURVEY

#### Existing Systems:

1. **CloudSpot: **

Hacker Rank is a site for Programmers from all over the world to solve programming problems in different CS domains like algorithms, machine learning and artificial intelligence, and to excel in different programming paradigms like functional programming. If you are not used to solving different kinds of problems, then it will appear hard. Hacker rank has around 5 levels of difficulty: Easy, Intermediate, Hard, Expert and Advanced.

#### LeetCode:

LeetCode is one of the most well-known online judge platforms that you can use to practice your programming skills by solving coding questions. It has over 1,100 different problems, support for over 18 programming languages, and an active community that is always there to help you with the solutions you come up with. If your intention is to hone your coding skills, then this online judge platform is one of the best that you can use.

#### CodeWars:

CodeWars provides a large collection of coding challenges submitted and edited by their own community. You can solve the challenges directly online in their editor in one of several languages. You can view a discussion for each challenge as well as user solutions.

#### C:\Users\tejas\Downloads\flaw.pngFlaws in Existing System:

1. **Developer-Oriented**
2. **Costly**
3. **Frustrating**

After studying this existing system, it concluded that these sites are not suitable for beginners as these sites are developer-oriented rather than being student-oriented. People need to pay for solving premium problems, and thereby limiting the learning. After solving a few problems, the level of problems drastically changes and some people founds it frustrating.

#### Proposed System:

As a solution to flaws in existing system, we came up with *‘Can’tCode’* that is:

1. **Student-Oriented**
2. **Free & Open-Source**
3. **Consistent Difficulty of Problems**

*'Can'tCode'* provides a potential solution to the problems mentioned above. It is more of a student-oriented platform. The Existing platform provide a wide range of programs to practice, but they are more difficult and confusing as they are developer oriented. Whereas, using *‘Can'tCode’* can be very friendly to the beginners as it is free-of-cost. Also, the difficulty levels in *'Can'tCode'* increase gradually, helping the user to learn better at a suitable pace!



## REQUIREMENT ANALYSIS

#### Functional Requirements:

1. **Login:**

Admin logins to the system by entering valid user id and password which is already set in the database to upload the events and related information of events.

1. **Event Section:**

The system will provide event list of events from database. Short description and long description are provided with the image.

1. **Test cases result:**

The admin has to pass all the test cases(details required to upload event).

1. **Logout:**

After uploading event admin can logout of system.

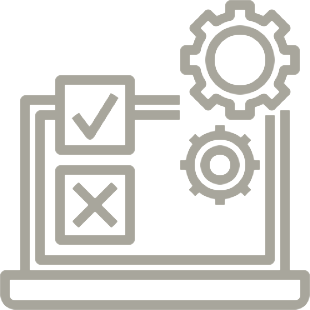
#### Non-Functional Requirements:

1. **Security:** The system’s back-end servers shall only be accessible to authenticated administrator. Sensitive data will be encrypted before being sent over internet.
2. **Reliability:** The system’s architecture is made having a focus on user attraction.
3. **Availability:** The system should be available at all times; the user can access it using a desktop or a laptop. It means 24x7 availability.
4. **Portability:** The application is a web app available and runnable on web browsers.

### Test Criteria:

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.** | **Functional Test Cases** | **Actual Output** | **Expected Output** |
| 1. | Verify if a admin is able to login with appropriate credentials. | Admin is logged in with system. | Admin logged in successfully |
| 2. | Verify if clicking on login button will redirect to admin panel. | Clicking on particular link is redirecting to proper page | Redirection on proper page |
| 3. | Verify if user is able to access about us, user section, contact us page. | User can access all the three pages | Accessing all three pages |
| 4. | Clicking on Explore our events is redirecting to the event section page where user is able to view events uploaded by admin. | User can view events uploaded by admin. | Viewing events uploaded by admin. |
| 5. | Clicking on DELETE and EDIT button events should be deleted and updated respectively. | Deleting and updating events is possible. | Successfully deleting and updating events. |

**Table. 3.1: Test Criteria Table**



## DESIGN

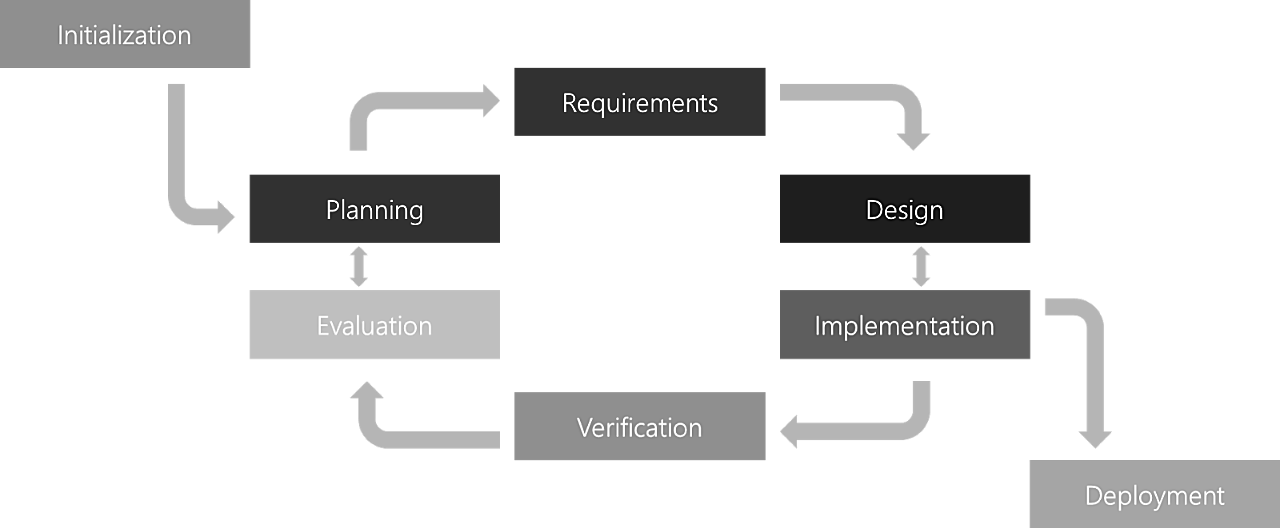
#### Software Development Life-Cycle (SDLC) Model:

The SDLC Model on which ‘IOPE Diaries’ is based on is ‘Iterative Model’.

#### Iterative Model:

Iterative process starts with a simple implementation of a subset of the software requirements and iteratively enhances the evolving versions until the full system is implemented. At each iteration, design modifications are made and new functional capabilities are added. The basic idea behind this method is to develop a system through repeated cycles (iterative) and in smaller portions at a time (incremental).

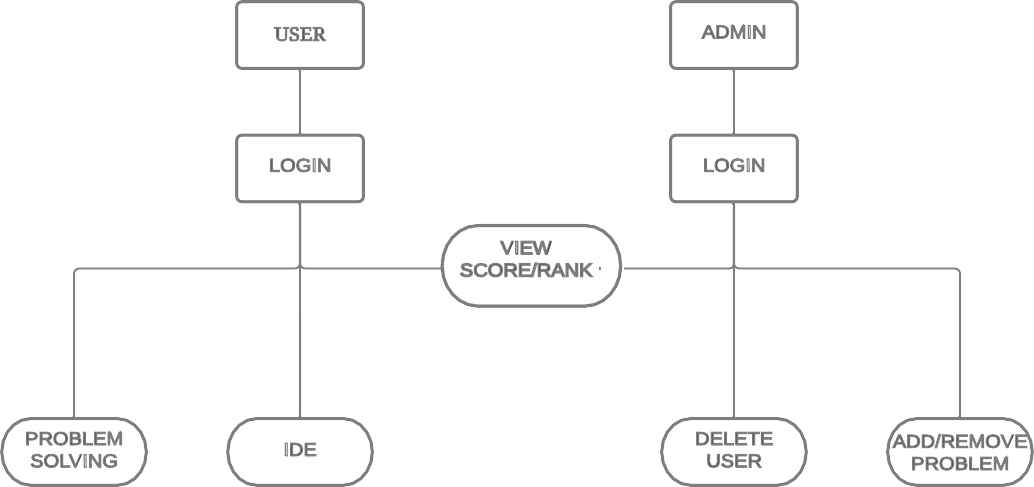
The following illustration is a representation of the Iterative and Incremental model −



**Fig. 4.1: Iterative Model (SDLC)**

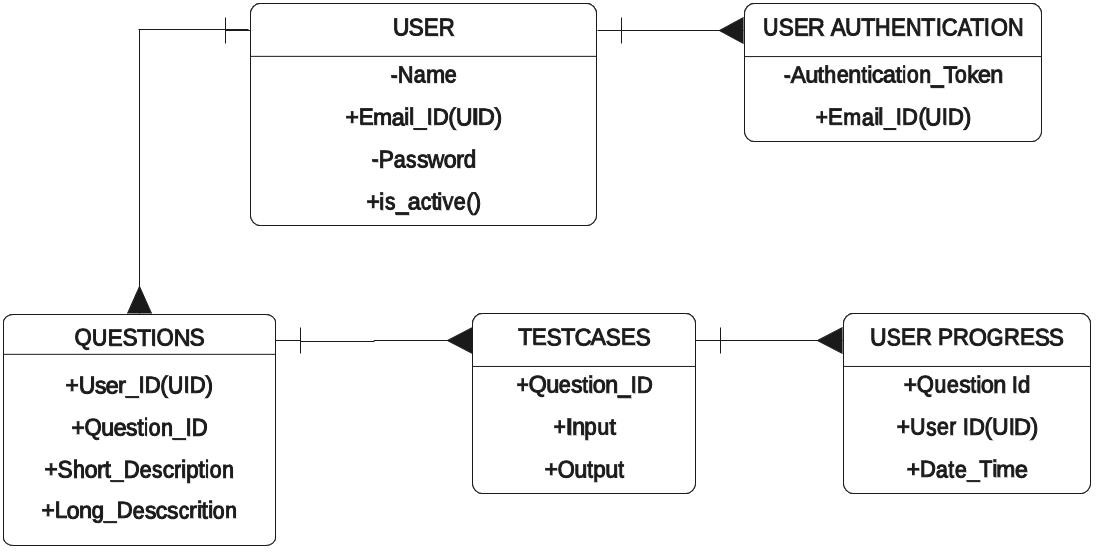
#### Flow Charts:

1. **Data Flow Diagram:**



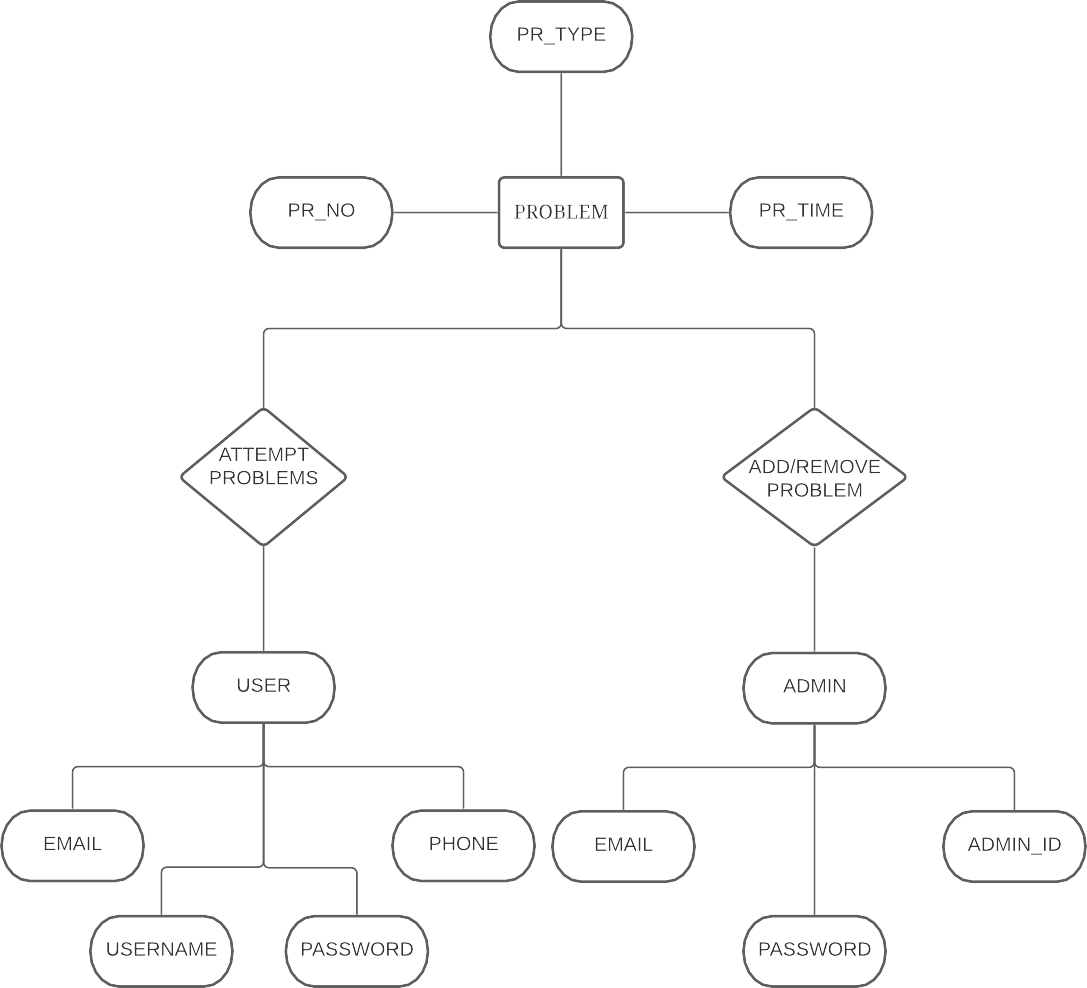
**Fig. 4.2: Data Flow Diagram**

#### Class Diagram:



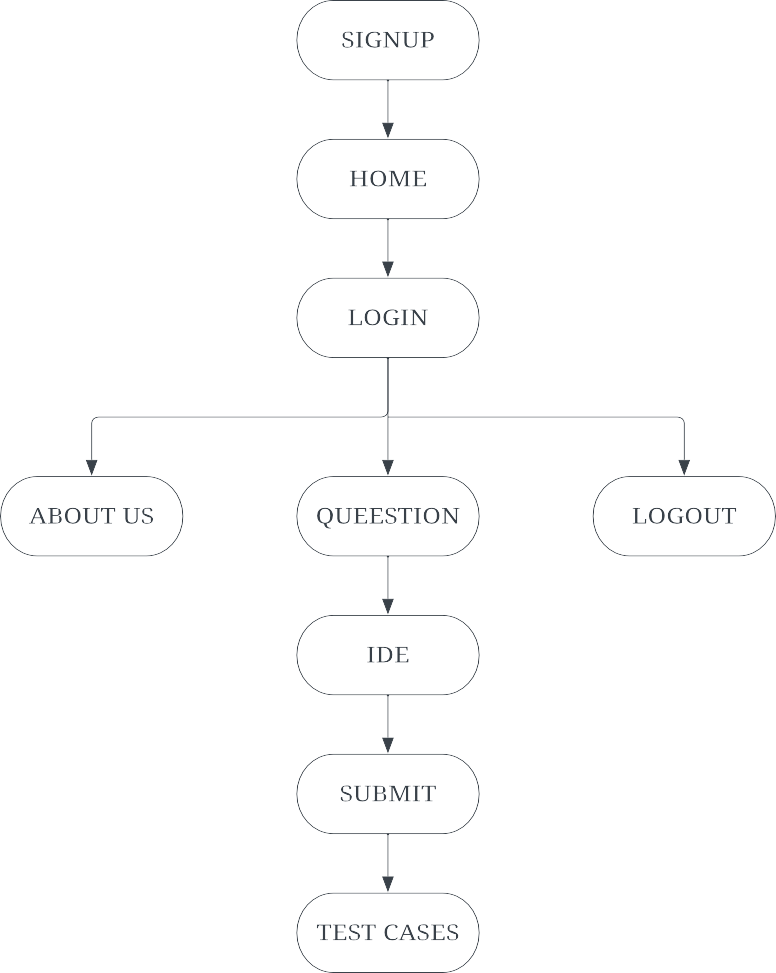
**Fig. 4.3: Class Diagram**

#### Entity-Relationship Diagram:



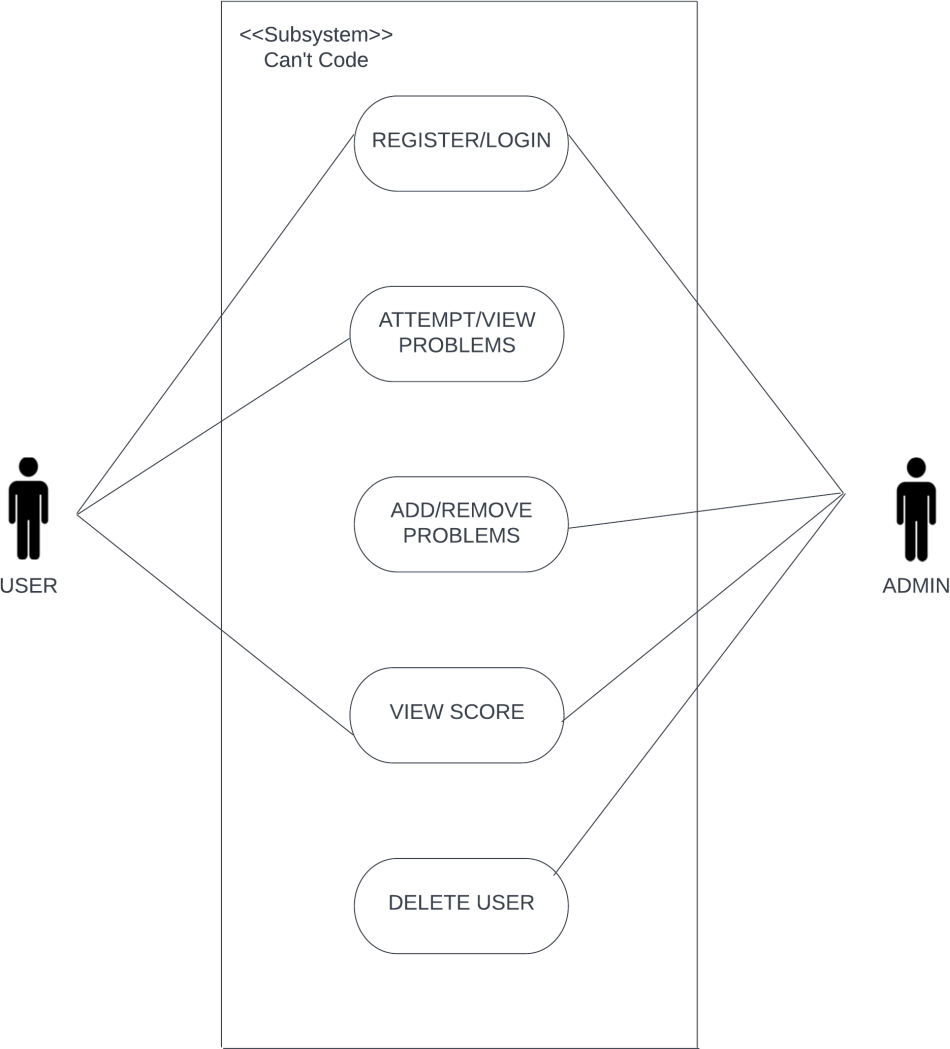
**Fig. 4.4: Entity-Relationship Diagram**

#### Control-Flow Diagram:



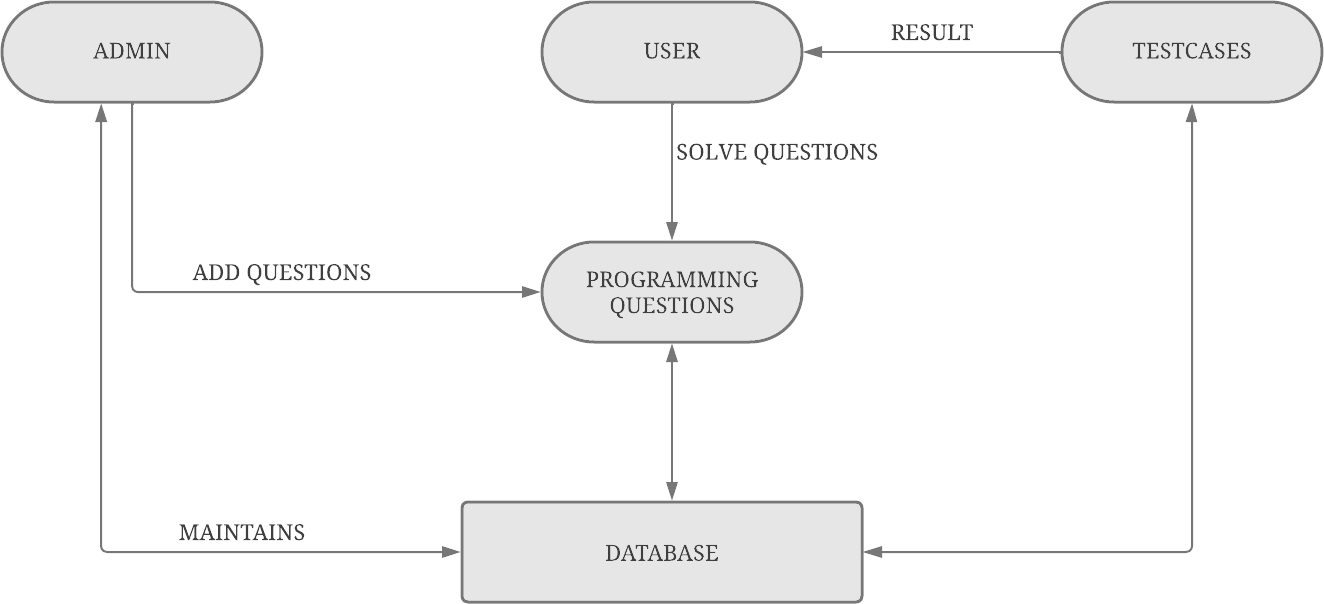
**Fig. 4.5: Control-Flow Diagram**

#### Use-Case Diagram:

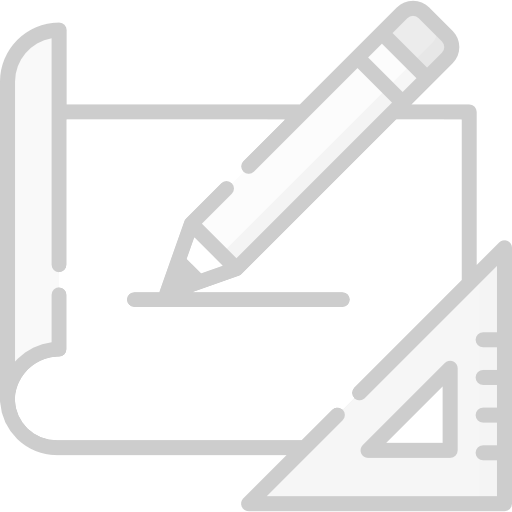


**Fig. 4.6: Use-Case Diagram**

#### System Architecture:



**Fig. 4.7: System Architecture**



## CONCLUSION

## REFERENCES