OOAD MINI PROJECT

CODE MENTOR

INTERACTIVE PROGRAMMING LANGUAGE LEARNER

KALAIDHARUN K (2022503009)
MALARVANNAN M (2022503011)

Table of Contents

1) Introduction.....4

2) Problem Statement.....4

3) Solution for problem statement.....4

4) Product description.....4

5) Intendend Audience.....5

6) Requirements.....5

6.1) Functional Requirements.....5

6.2) Non-Functional Requirements.....6

7) UML diagram 7

7.1) Use case diagram.....8

7.2) Class diagram.....7

7.3) Module diagram.....9

7.4) Class-Responsibility-Collaboration Cards.....10

7.5) Data flow diagram.....12

7.6) Sequence diagram 15

7.7) State Machine diagram 15

8)Project Demonstration.....16

8.1) Login Page.....17

8.2) Main Page.....17

8.3) Learning Hub 18

8.4) Learn C and Java.....19

8.5) Test Page.....19

8.6) Daily Page.....20

8.7) Settings.....20

8.8) Profile.....21

9)Technology Stack Used.....21

10) Single Module Test.....22

11) Conclusion.....22

1.Introduction:

In today's digital age, coding has become an indispensable skill, yet traditional learning methods often make the journey, challenging especially for beginners. The complexities of syntax, the dryness of theoretical lessons, and the lack of immediate feedback can be discouraging. Recognizing these challenges, we introduce the Interactive Programming Language Learner (IPLL)

2.Problem Statement:

1. In the rapidly evolving tech landscape ,learning programming languages effectively and engagingly remains a significant challenge.
2. Traditional education methods often fall short in providing interactive and practical learning experiences.
3. The "Interactive Programming Language Learner" project aims to bridge the gap by creating a dynamic platform that revolutionizes how individuals learn programming languages.

3.Solution for Problem Statement:

The IPLL platform aims to track and manage various types of user data, such as progress on learning modules, quiz results, badges earned, and daily question engagement. The platform also needs to store a diverse set of programming challenges and associated metadata, including difficulty level, language type, and user interaction records (e.g., attempts, feedback). Traditional SQL-based databases, while effective, may face challenges in scalability and flexibility as the amount and variety of data increase.

4.Product Description:

In the rapidly evolving tech landscape, learning programming languages effectively and engagingly remains a significant challenge.

The "Interactive programming Language Leaner" project aims to bridge this gap by creating a dynamic platform that revolutionizes how individuals learn programming languages.

By integrating interactive coding challenges and feedback, this platform will transform the learning journey into an engaging and productive experience, equipping learners with the skills they need to excel in the programming world.

IPLL is intended to replace the dry learning environment of Programming languages with an interactive with an interactive, playful, and colorful environment.

5.Intended Audience:

Our product will be very useful for beginners, early coders, active learners, and passionate coders.

6.Requirements:

6.1 Functional Requirements:

1. **Providing Choosing the programming language.**
    a. We will be providing the user an option to choose the programming language he wish to learn.
    b. By this option the user can choose his desired language and start to embark his coding journey.
2. **To allow user to engage in activities like learn and attend test**
    a. The user can either learn a language or he can test his knowledge he posses in the programming language.
    b. In learn we will be teaching the user basic syntax and the path of the programming language
    c. In test section the user will be able to attend test based on his progress level. With this progress level the user will be prompted with the difficulty of the problem
3. **Provide a interactive session.**
    a. The learning process will be very much interactive as we would try to gamify the learning process, making it colorful, interactive, visually appealing.
4. **Teach user from basic to advance concept.**
    a. We would teach the user from the very basic syntax and coding modules, as he progress through our learning process, we would teach him advance components.
5. **To conduct small quiz**
    a. There's no use in just learning. One would constantly need to test his might in order to excel in his field.
    b. With the above mentioned point on point, we will be conducting small quiz at the end of every lesson.

c. .

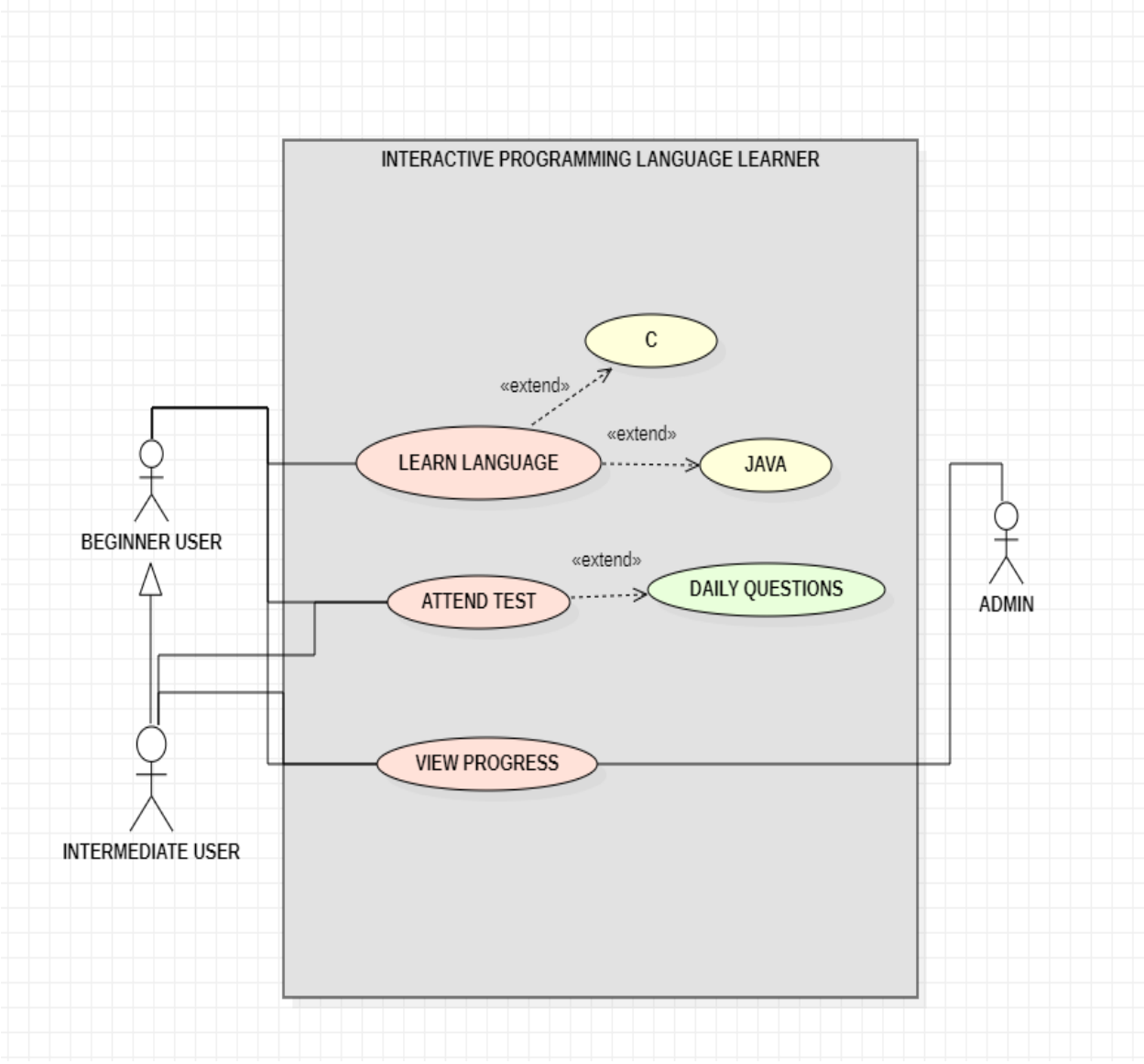
6. **Provide daily question**
    a. We will be using any API or we will manually integrate an option to display the user daily questions. With this the user will be able to increase his knowledge as he practice regularly.
7. **To check overall progress**
    a. The user will be able to check his progress, both learning and test.
8. **To provide Badges**
    a. Based on the progress the user will be given badges which would be helpful for him to stay motivated..

6.2 Non functionality Requirements:

1. **Usability (user friendly interface)**
    a. The interface should be intuitive, making navigation easy for users of all skill levels.
    b. Consistent layout and design patterns will improve the user experience.
2. **Performance (Fast loading times)**
    a. The platform should respond quickly.
    b. Efficient use of resources, minimizing unnecessary server requests and optimizing code.
3. **Reliability**
    a. The system should have minimal downtime, aiming for high availability.
    b. Fault tolerance mechanism
4. **Portability.**
    a. The code should be easily adaptable to different environments.

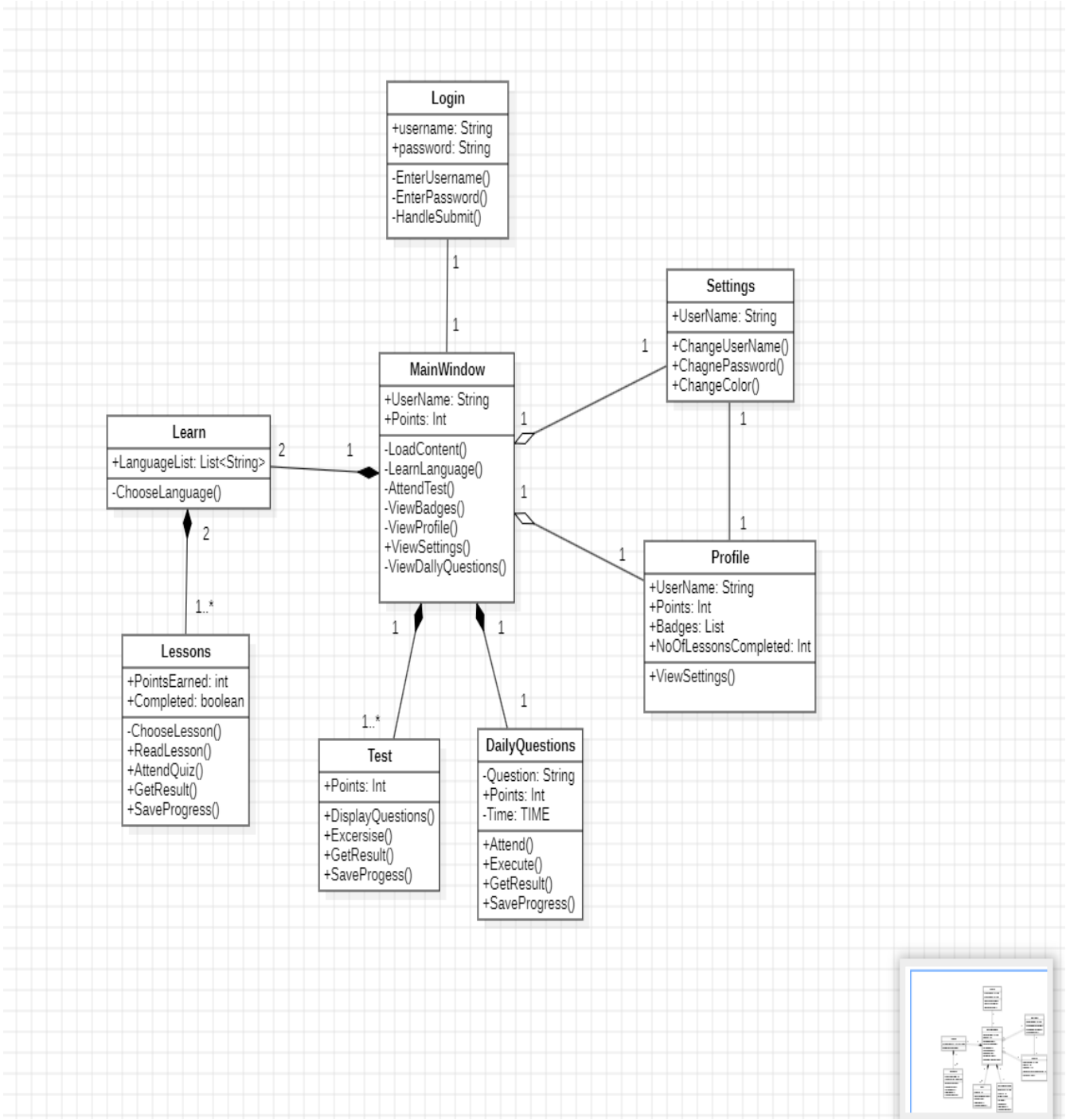
7.UML diagrams:

7.1 Use case Diagram



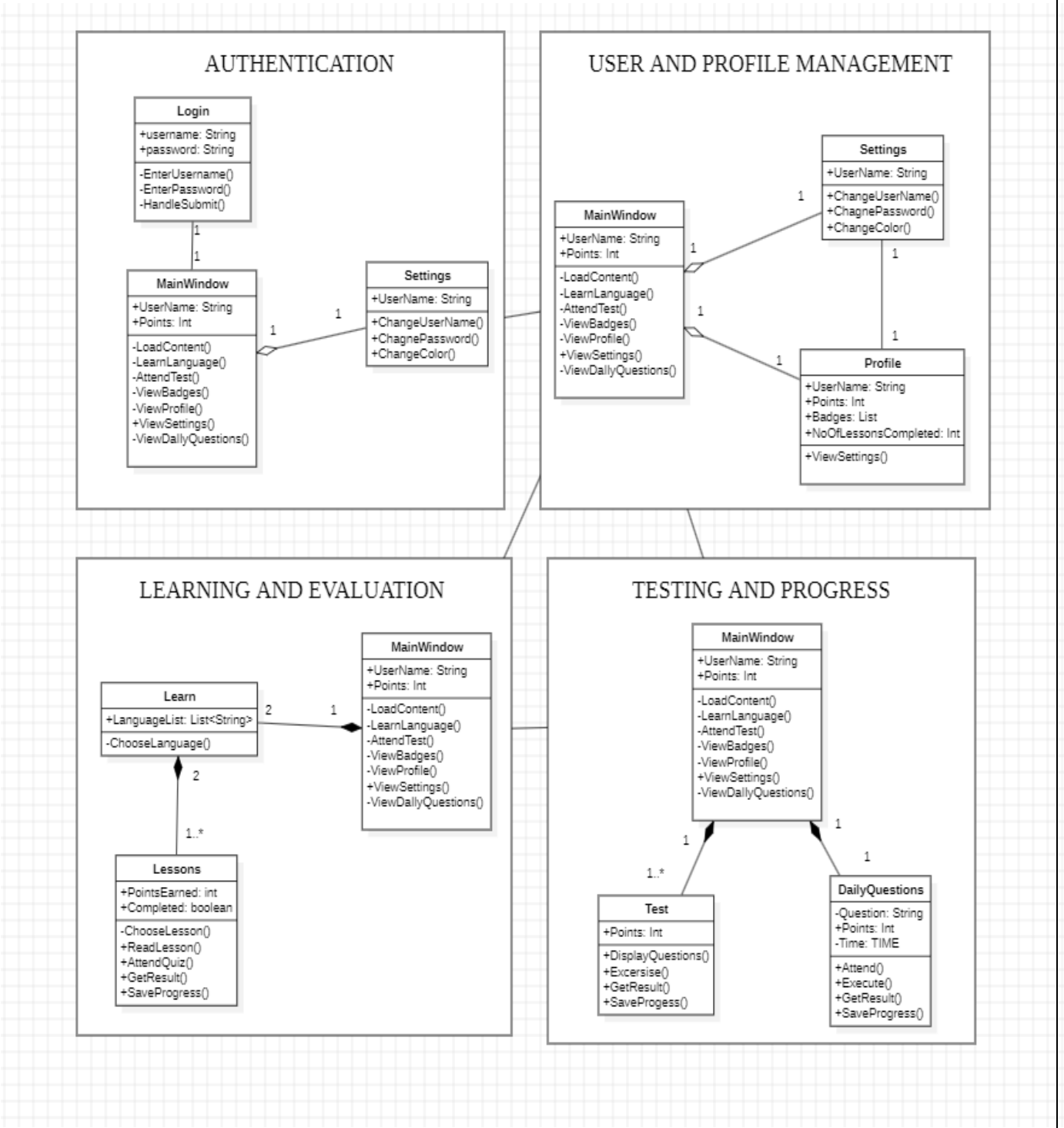
The above diagram depicts the overall use case of the project.

7.2 Class Diagram



The above diagram depicts the overall Class Diagram of the project.

7.3 Module Diagram



The above diagram depicts the overall Module components.

7.4 Class-Responsibility-Collaboration Cards

LOGIN	
- Get username and password from user - Validate the credentials - Redirect to the main window after getting the valid credentials	- MainWindow

MainWindow	
- Load the content on valid credential login - Provide buttons to learn, test their knowledge, and provide daily questions - DIsplay points of the user. - Allow user to navigate to Settings - Allow user to see his profile - Display the badges	- Login - Settings - Profile - Learn

Settings	
- Allow user to change username and password - Allow user to change the color of the highlighter	- MainWindow - Profile

Profile	
- Check the user profile. - Allow user to check the username. - Display points - Allow user to view the settings	- MainWindow - Settings

Learn	
- Allow user to select the language, either C or java learning path. - Connects with the main window, navigate between them.	- MainWindow - Lessons

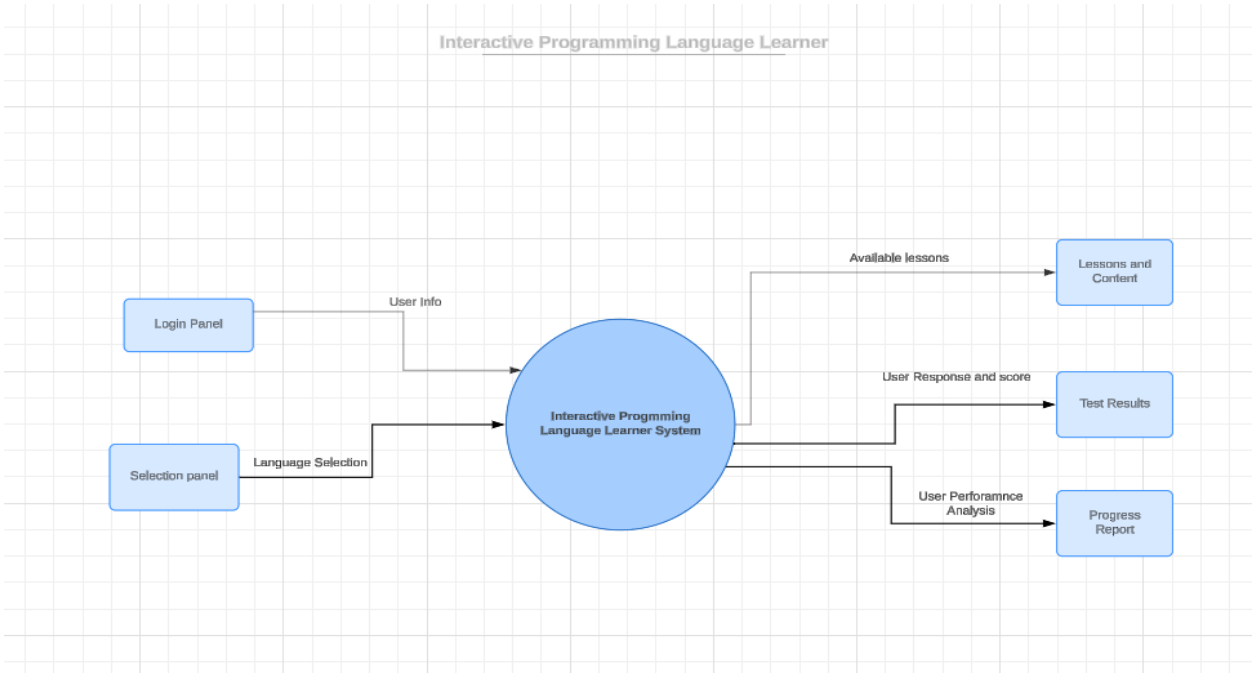
Lessons	
- **Throw user lessons to learn** - **Allow user to attend quiz at the end of each lesson** - **Unlock next lesson only when you complete current lesson** - **At the end of quiz add the points and give result to the user** - **Save the progress automatically and update the points**	- **Learn**

Test	
- **Allow user to attend the test.** - **Display questions based on progress level** - **Save progress on every test completion and update the points** - **Get result and display to the user**	- **MainWindow**

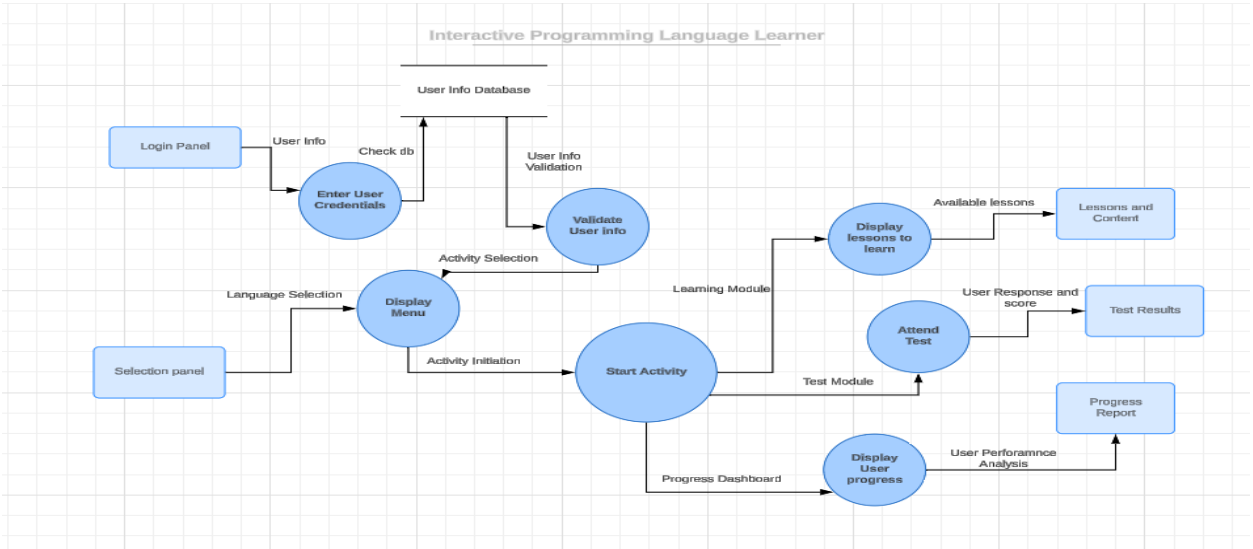
DailyQuestions	
- **Provide questions that resets for every 24 hrs** - **Allow user to attend the test.** - **Save progress and update the points.** - **Get result and display to the user**	- **MainWindow**

7.5 Data Flow Diagram

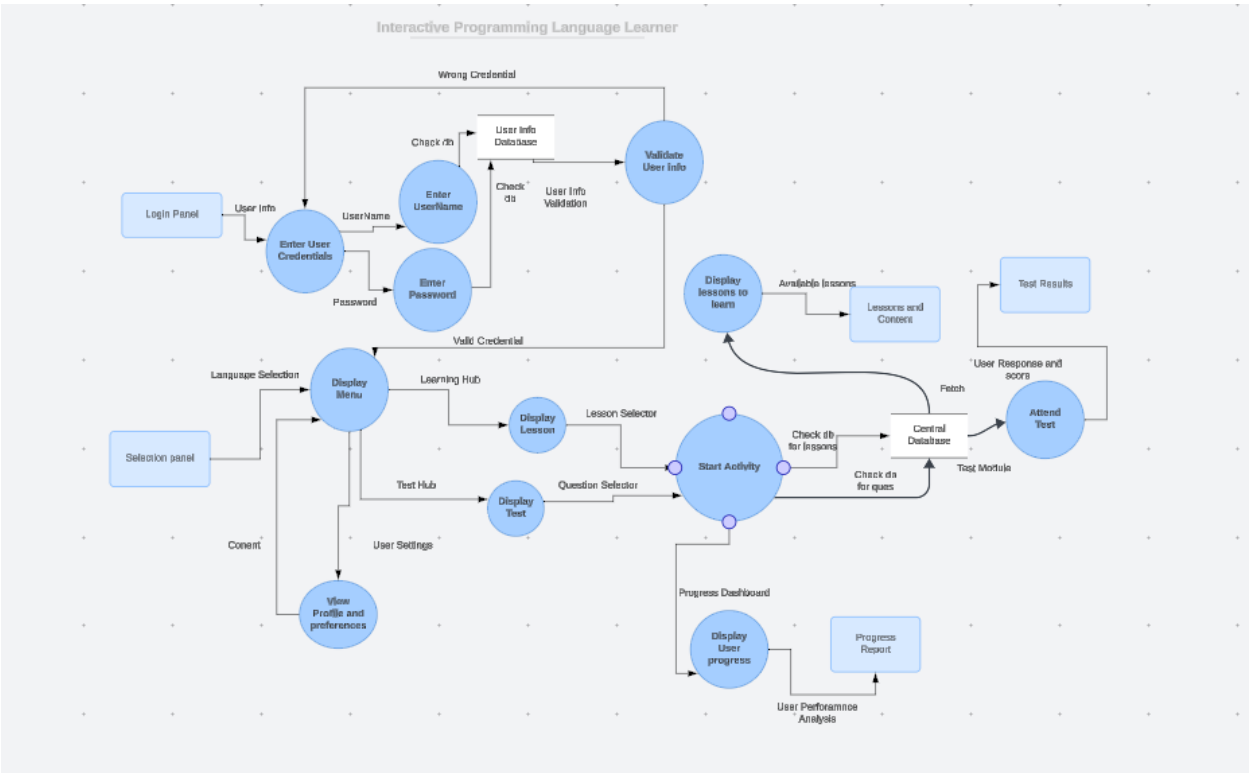
Level 0:



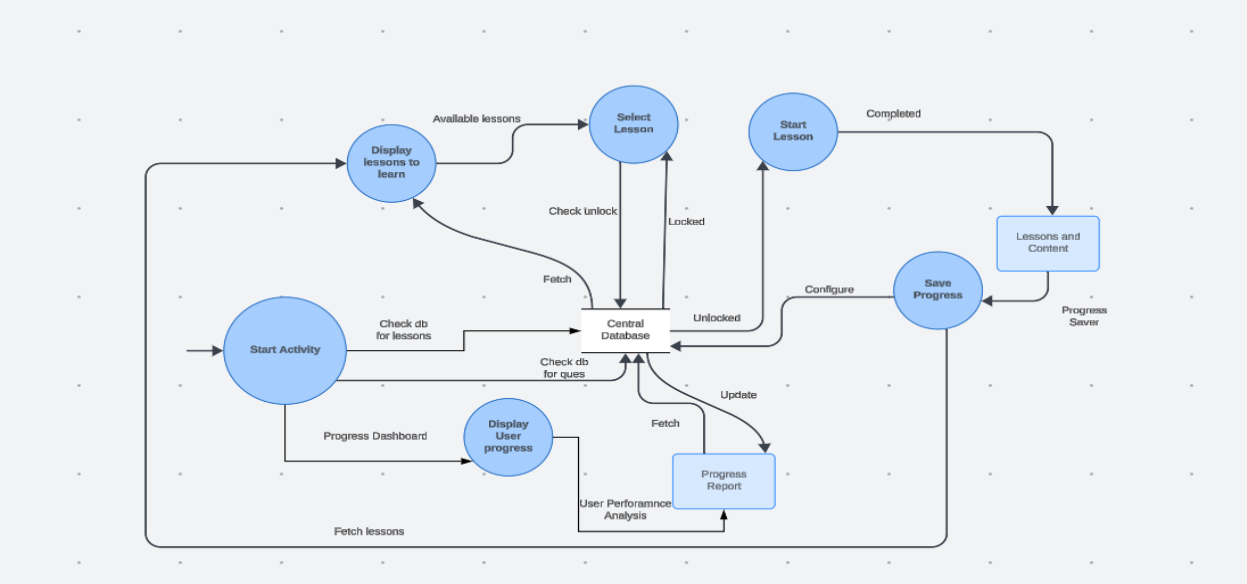
Level 1:



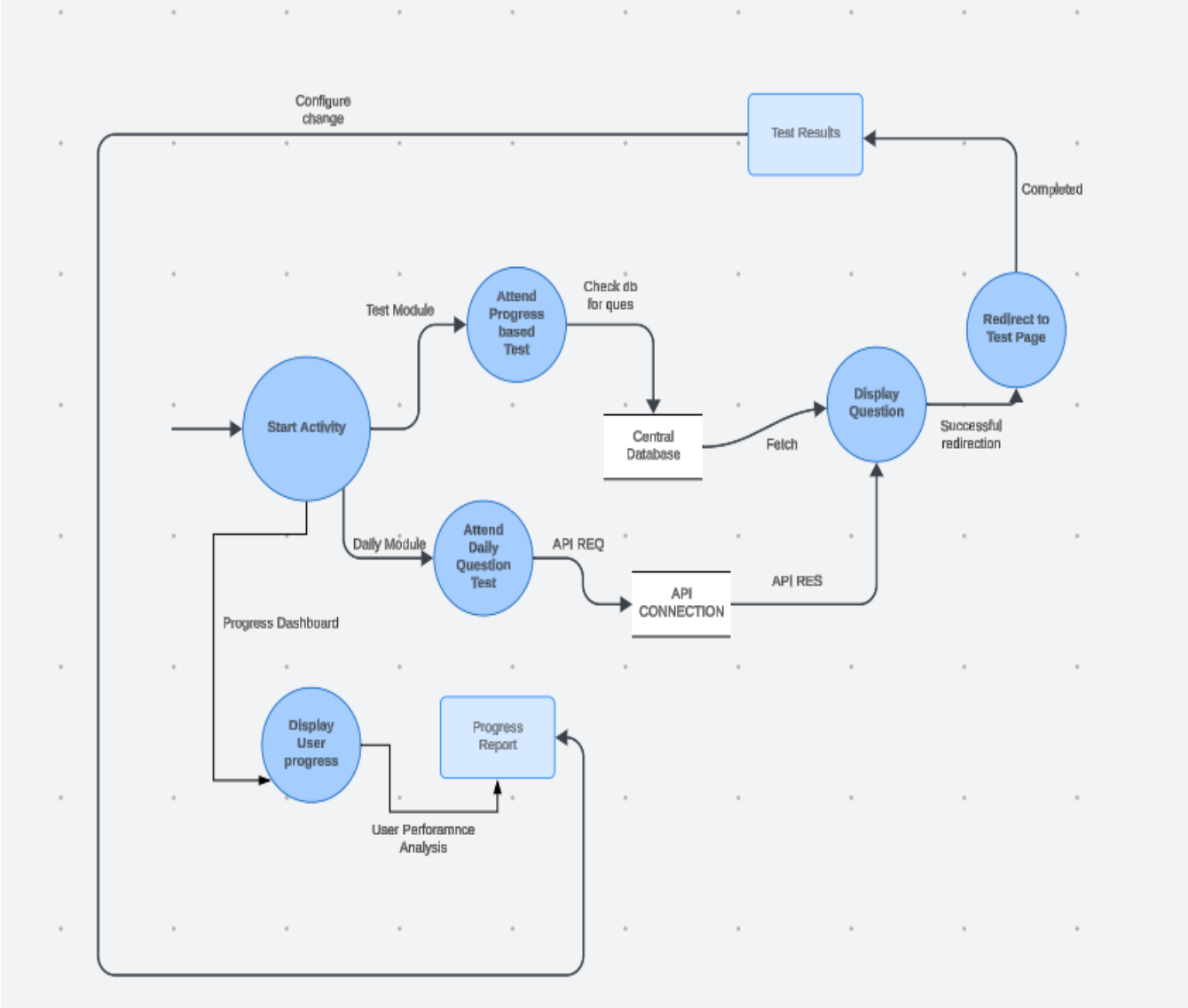
Level 2:



Level 3:



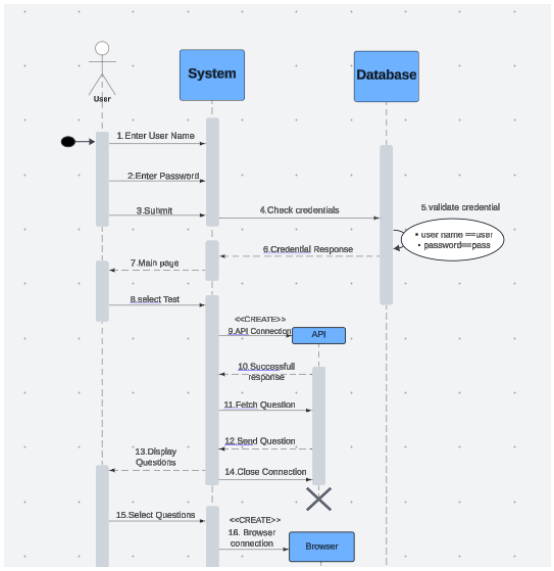
Level 4:



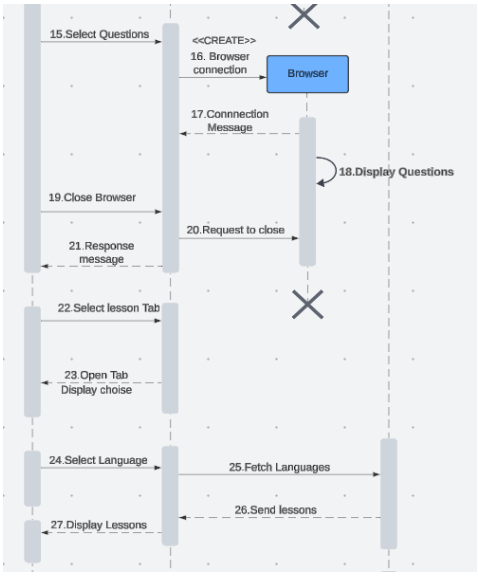
7.6 Sequence Diagram

This sequence diagram show the interaction between system, database, API and user with each other starting from login page then moving to hub and then to lesson and how the database is fetched and updated

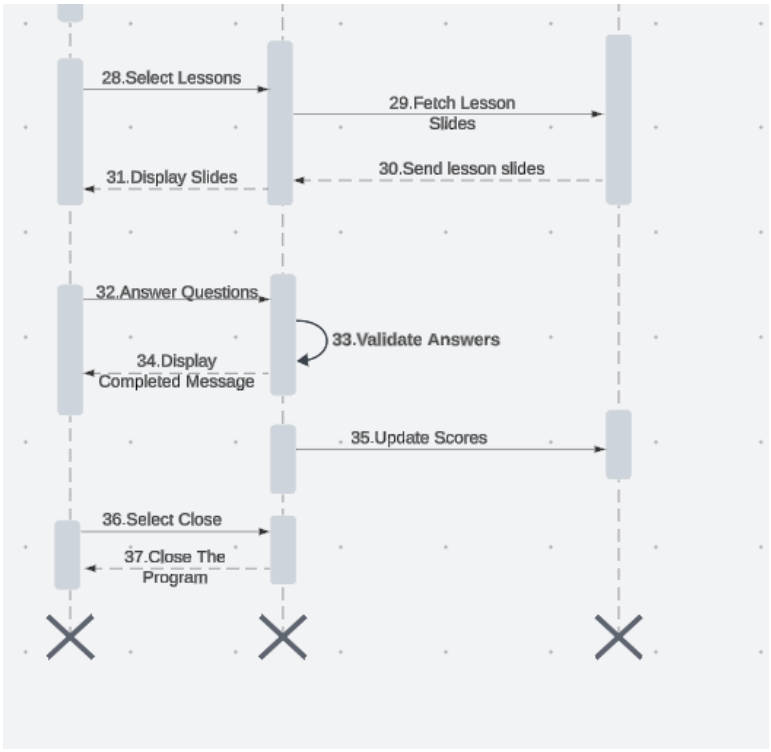
A.1



A.2

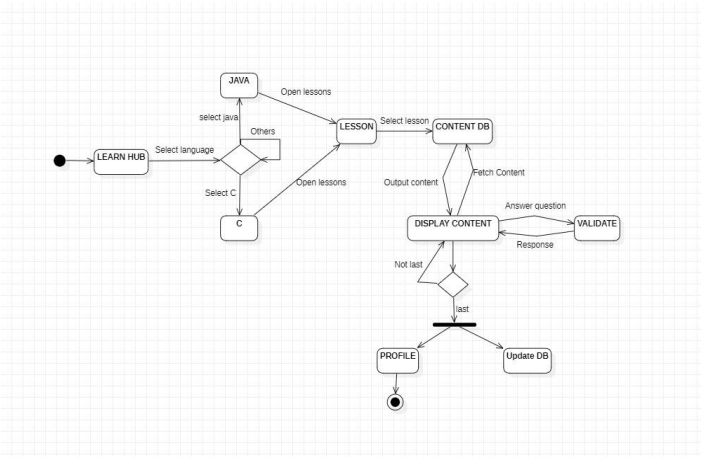


A.3



7.7 State Machine Diagram

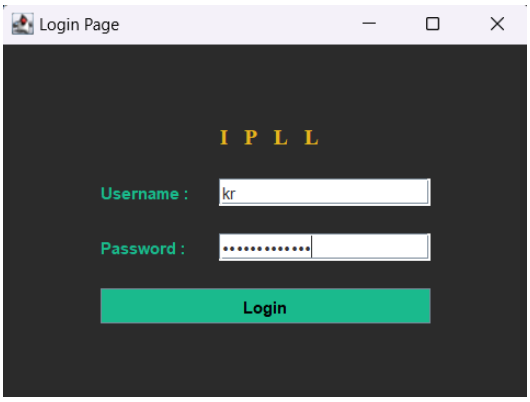
This State machine diagram depicts the flow of states between learning hub and lessons



8. Project Demonstration:

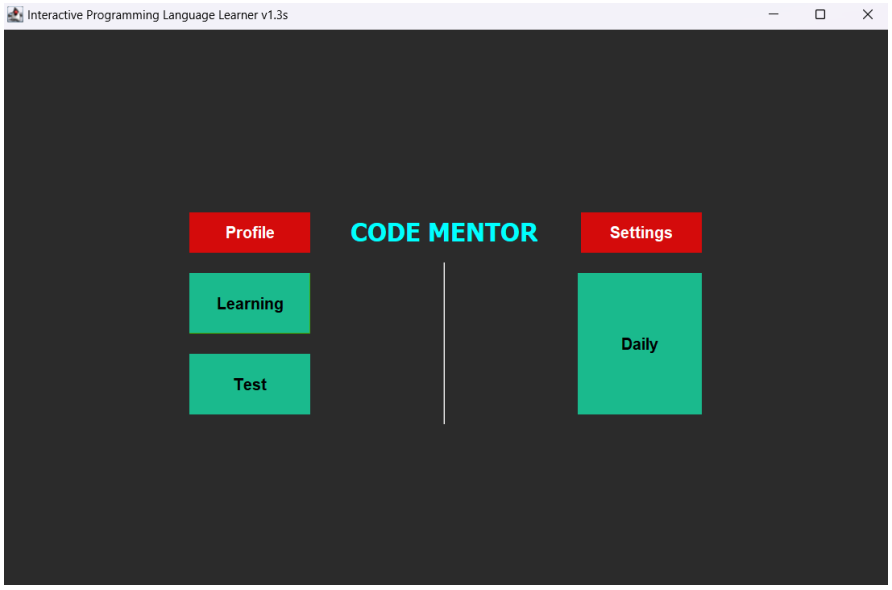
8.1 Login page:

The user (single user) is asked with username and password. It searches the database (mongo) for username and password, if it is found then it moves the next page Leaning Hub, otherwise the user is popped with a dialog box



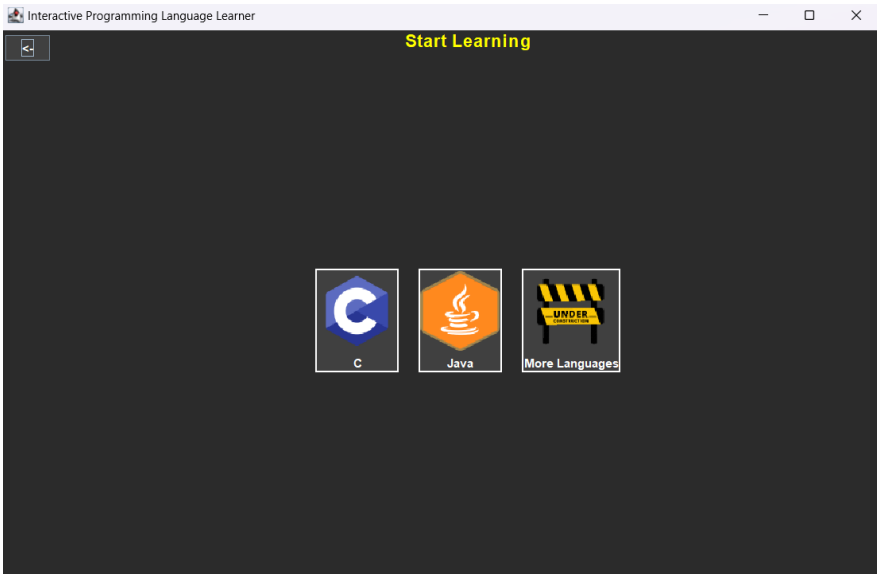
8.2 Main page:

Our main page consist of options to move to settings, profile page, learn hub, test page where number of questions are displayed using CodeForces API, and Daily test.



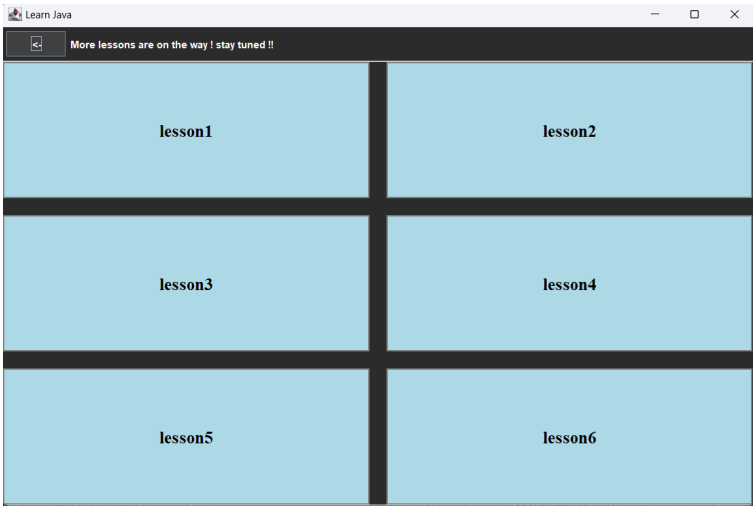
8.3 Learning Hub:

In this page the user can select the language he want to learn. For now we have added Java and C. And in future we may include more languages



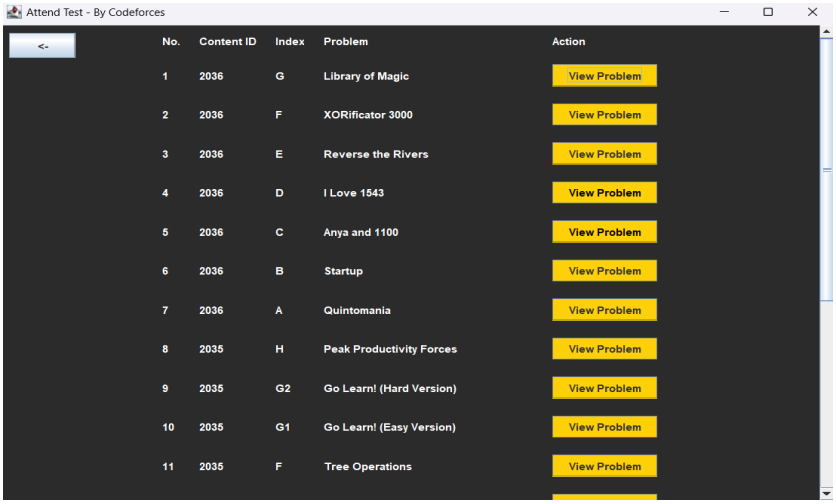
8.4 Learn C and Learn Java:

Here the user can select the lesson. The user can select the lesson only if the previous of the selected lesson is completed. If a lesson is completed then the score will be uploaded in the database. The contents are loaded from the database



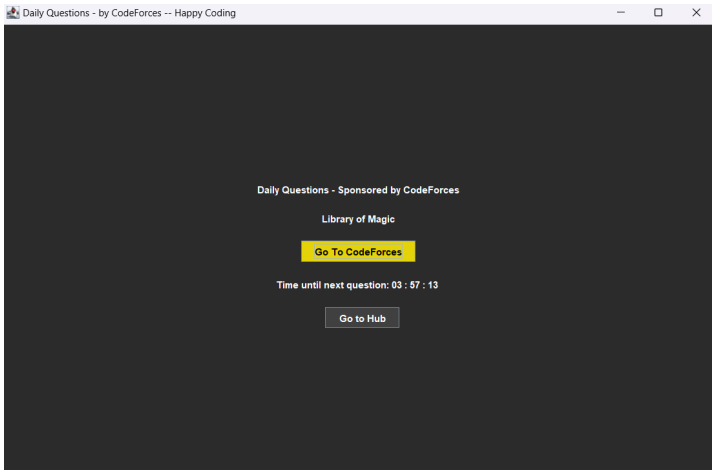
8.5 Test page

With the help of the CodeForce API we have fetched questions and displayed in our frame. If we click the button associated with it, it will move to the browser which has the question



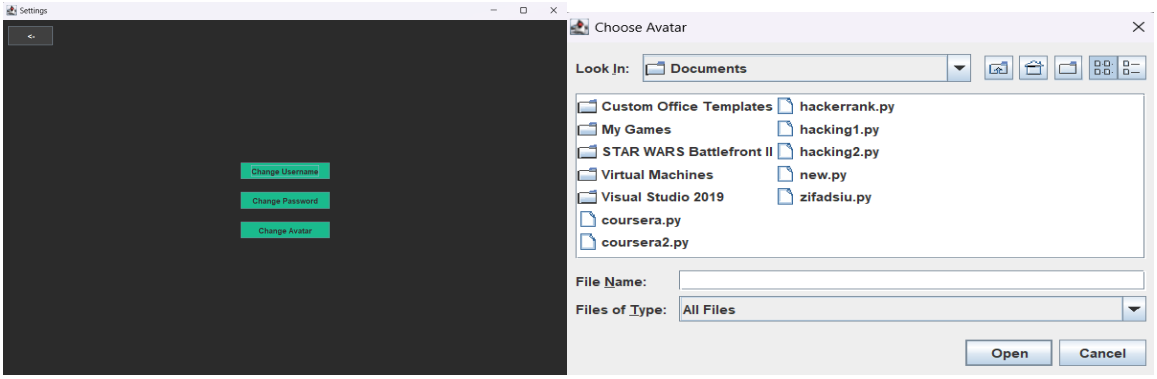
8.6 Daily page:

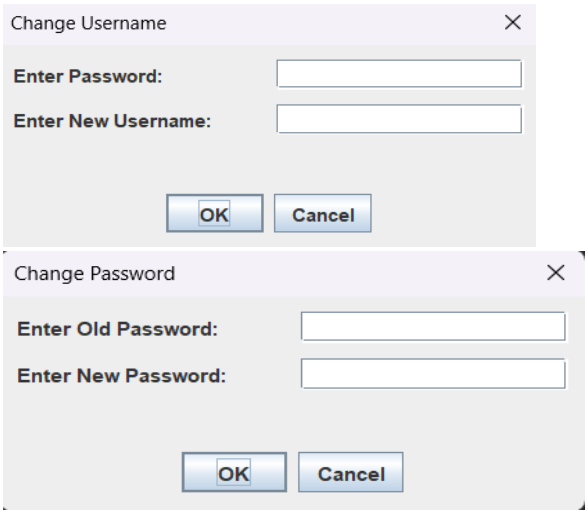
In this page with the help of CodeForce API, we have fetched a single question and it will automatically get updated for every 24 hrs.



8.7 Settings Page:

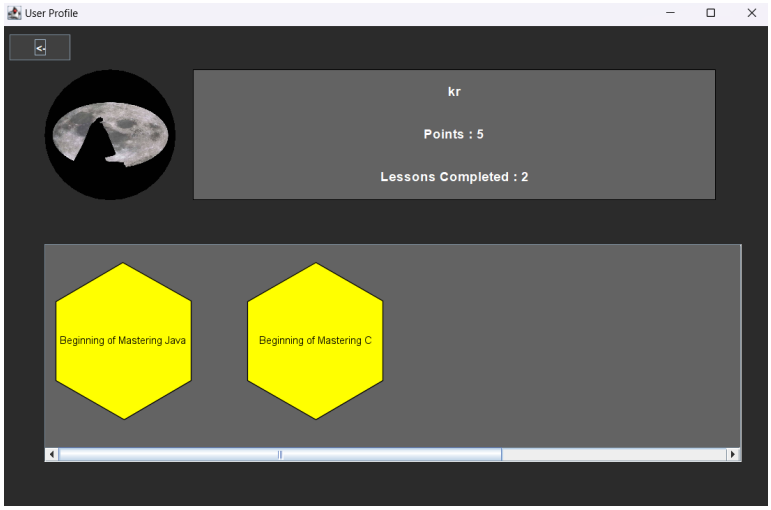
In Settings the user will be able to change his username, password , avatar





8.8 Profile:

In profile the user will be able to view his detail like name , lessons completed , profile pic and the badges he have earned



9.Technology Stack Used:

Java & swing:

Java swing is used as the major User interface building tool.Components like JLabel, JButton, JTextArea, JFrame, JPanel etc were widely used in this project.

Mongo DB:

With the help of external jar files (for json and mongo connectivity) we were able to connect to mongo db and we have successfully connected to the database, fetched collections, extracted documents and successfully updated the values in the collections.

We have created separate collections for each lesson

Codeforces API:

With the help of the codeforces API we have successfully fetched the questions and displayed in our frame

It was useful in both test section and as well as Daily section

10.Single Module Test:

Module Test have been conducted for the **main page** and the result is pasted below

11.Conclusion:

The **Interactive Programming Language Learner (IPLL)** platform addresses the limitations of traditional programming education by creating an immersive, engaging learning environment. IPLL transforms coding into an enjoyable and productive experience. IPLL not only supports beginners in grasping foundational concepts but also provides continuous challenges to maintain their learning trajectory.