

**LiveCode Compiler**

**School of Computer Science & Engineering**

**Academic Year 2022-23**

**Mini Project on Open-Source Development**

**Synopsis**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr No.** | **Name** | **PRN** | **Mobile No.** |
| **1** | Sonika Varma | 1132220233 | 9168433666 |
| **2** | Rohan Dhanani | 1132220432 | 9510578564 |

**Project Overview :-**

**Introduction :-**

* In this project, we created a real-time code-editor or compiler. In a typical compilation, a user can simply code and see; however, in our project, a user may establish a room with socket.io, invite his friends in, and then compile and run the code together. Additionally, a user can teach the code using the room capability.

**Need of Work / Reason for selection of this project :-**

* The main motivation behind creating this project was to address the fact that, in a typical compiler, only one person could write and run the code while another could just observe. However, with this project, anyone could create a live room and invite his friends to the room. and all working together to compile, run, and monitor the code. In this project, you can code in the language of your choice.

**Objectives :-**

* Create a Web real-time compiler that allows you to create a live room, add a friend, and collaborate. Additionally, you can use this project as a teaching tool to show students how to write and run code using a compiler.

**Problem Statement:-**

* A real-time code editor refers to an application that enables multiple users to collaboratively edit and write code simultaneously. The concept involves providing a shared virtual environment where programmers can work together on a coding project, allowing real-time updates to be visible to all participants. This collaborative approach fosters efficient teamwork, making it suitable for pair programming, code reviews, and remote collaboration.
* One notable implementation of a real-time code editor is outlined in a research paper titled "Real-time Code Editor Application for Collaborative ..." which analyzes the benefits of such tools for cooperative coding projects. This approach enhances productivity, as it eliminates the need for developers to manually share code files and integrates communication features for seamless interaction.

**Flow Of System :-**

## **One Entities:**

* 1. Client/User.

## **User/Client Side:**

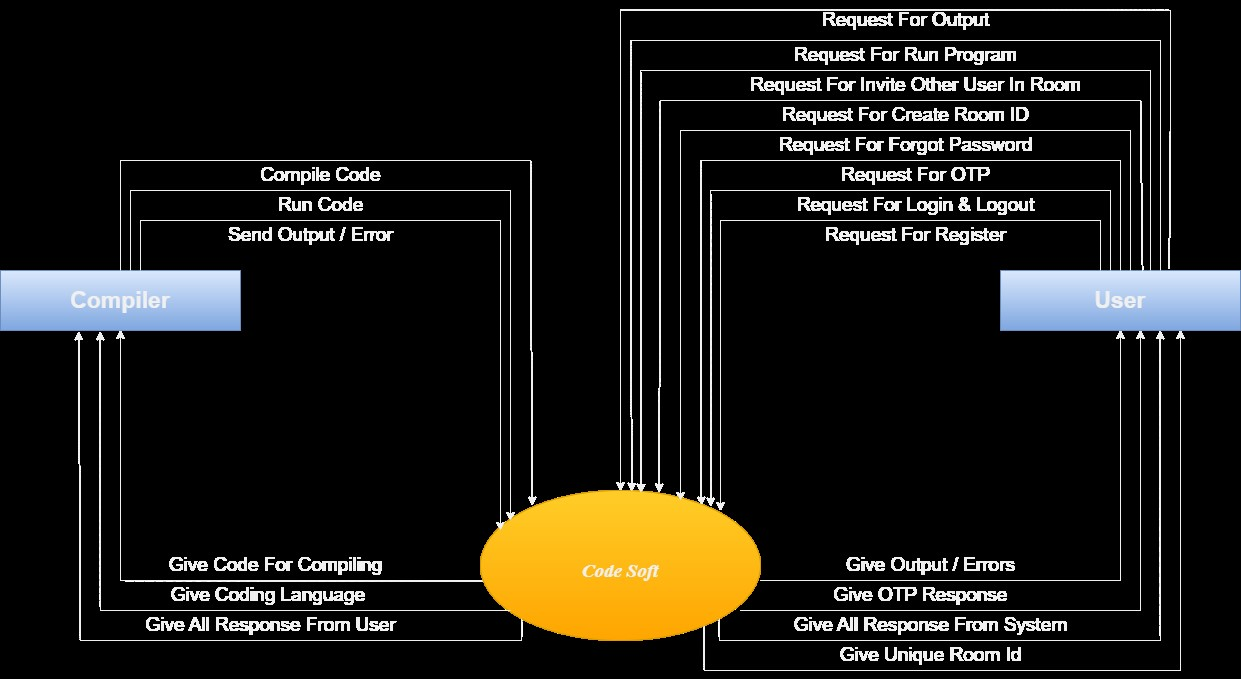
1. Sign Up.
2. Sign In.
3. Forget Password.
4. Edit Profile.
5. Create Room.
6. Invite Users.
7. Write Code.
8. Run Code.
9. Give Input.
10. Get Output.
11. Feedback.
12. About Us.
13. Contact Us.
14. Logout.

## **Compiler Side:**

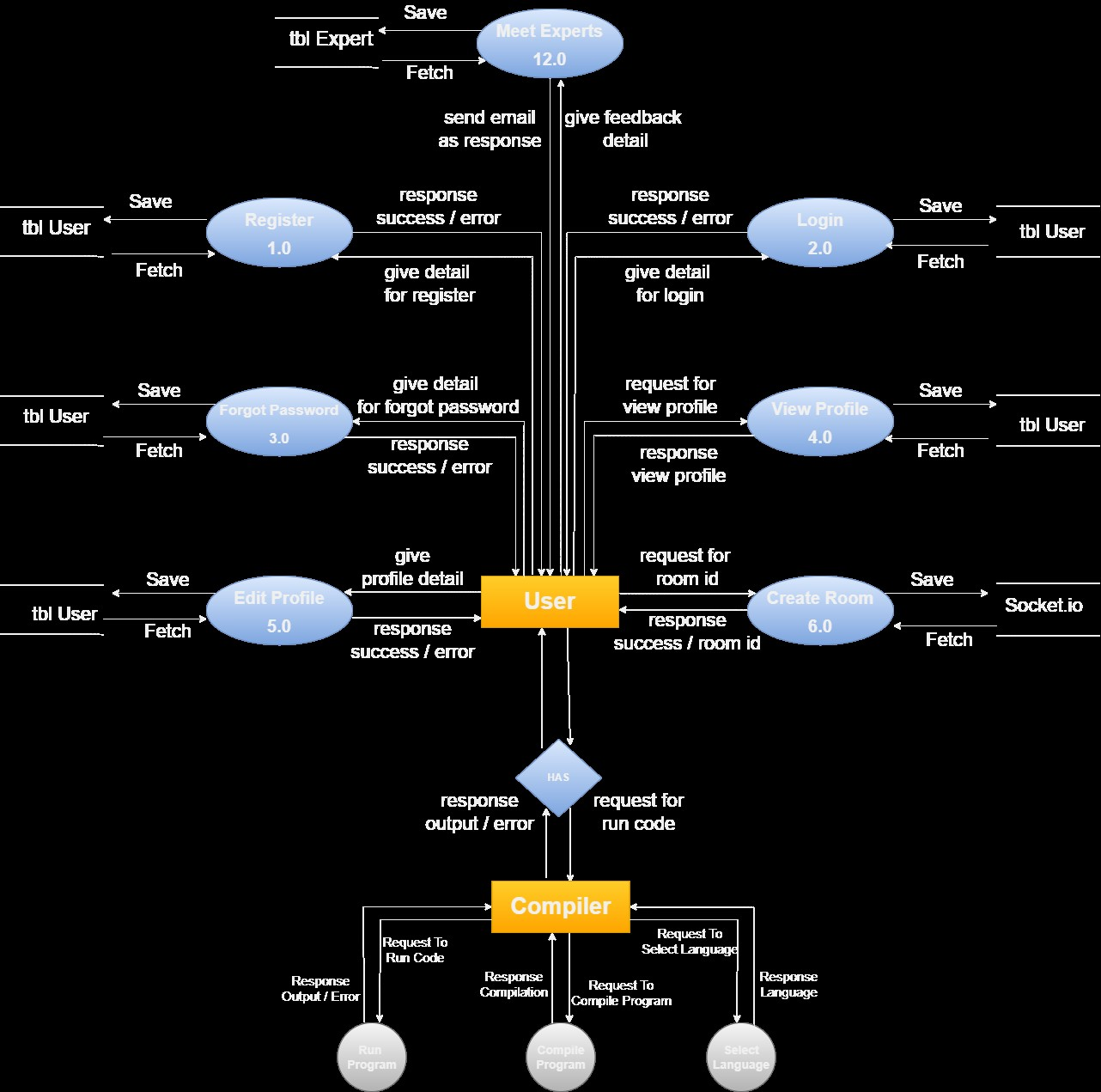
1. Create Room.
2. Send Room Invitation.
3. Select Language.
4. Compile Program.
5. Run Program.
6. Take Input.
7. Give Output / Error.

**Data Flow Diagram :-**

**0 level DFD :-**



**1 Level DFD :-**



**Benefits to the surrounding/society :-**

* As a result of the COVID-19, schools all throughout the world have been closed. Over 1.2 billion youngsters are not in school worldwide.
* As a result, education has undergone a significant transformation due to the advent of e-learning, in which lessons are delivered online and through digital platforms.

**Advantages :-**

* Real-time compiler facilitates anywhere learning and coding.
* Lectures can be taken any number of times.
* Web real-time compiler that allows you to create a live room, add a friend, and collaborate.
* Real-time compiler Learning is cost effective as compared to traditional forms of learning.

**Limitations** :-

* Not available as of yet in mobile application form.
* Network connectivity is required.
* This project does not have chat or video call features.

**Project requirements :-**

* Equipment: Broadband, Laptop/PC
* Facilities required (include software, hardware):

1. Software :- Node.js, React.js, Css, Socket.io

2. Hardware :- Windows 10, Internet Connection

* Communication, Budget (if any) etc: