A

PROJECT REPORT ON

**DCODER**

By

**NIKHIL BAMBHROLIYA (CE-013) (19CEUEG099)**

**FENIL DOMADIYA (CE-037) (19CEUEG037)**

**JENIL GAJERA (CE-040) (19CEUET013)**

B.Tech CE Semester-VI

Subject: System Design Practice

Guided by:

Prof.Pandav K. Patel

Assistant Professor

Dept. of Comp. Engg.



**Faculty of Technology**

**Department of Computer Engineering**

**Dharmsinh Desai University**



**Faculty of Technology**

**Department of Computer Engineering**

**Dharmsinh Desai University**

**CERTIFICATE**

This is to certify that the practical / term work carried out in the subject of **System Design Practice** and recorded in this journal is the bonafide work of

**NIKHIL BAMBHROLIYA (CE-013) (19CEUEG099)**

**FENIL DOMADIYA (CE-037) (19CEUEG037)**

**JENIL GAJERA (CE-040) (19CEUET013)**

of B.Tech semester **VI** in the branch of **Computer Engineering**

during the academic year **2021-2022**.

|  |  |
| --- | --- |
| Prof. Pandav K. Patel  Assistant Professor,  Dept. of Computer Engg.,  Faculty of Technology  Dharmsinh Desai University,Nadiad | Dr. C. K. Bhensdadia,  Head,  Dept. of Computer Engg.,  Faculty of Technology  Dharmsinh Desai University, Nadiad |
|
|
|
|

# Table of Contents

# 

[**Table of Contents**](#_fcomp5krkx47)3

[**1. Abstract**](#_p35guzids5mv)4

[**2. Introduction**](#_ufr07ym2ufa2)5

[**2.1 brief introduction**](#_m4nvv8v2tj1c)5

[**2.2 Tools / Technologies Used**](#_9dj8eagqnin0)6

[**3. Software Requirement Specifications**](#_6a8ptfnljgt)7

[**4. Design**](#_p8rzyu3o0hyg)10

[**4.1 Use Case Diagram**](#_pt92fdakwnb7)10

[**4.2 Class diagram**](#_1yy9tlvd1kvj)10

[**4.3 Activity diagram**](#_eotqp3xku7l2)12

[**4.4 ER diagram**](#_yc8n8z2ubfl3)13

[**4.5 Data dictionary**](#_b2961h2t9a7r)14

[**5. Implementation Detail**](#_bv5vkycj4wdp)18

[**6. Testing**](#_zhu42ayjjz8)20

[**7. Screenshots**](#_cuj4mg9lggfl)21

[**8. Conclusion**](#_q6sgtgwkus7w)25

[**9. Limitation and Future Extension**](#_y3wykm85ho0v)26

[**10. References**](#_kdfq05o9t6j7)26

# 

# 

# 1. Abstract

Decoder is a website that contains an online compiler for some of the languages. Users can read the questions and can compile, save and share their code. Also, Users can get the data regarding upcoming contests of various platforms like Codechef, Codeforces, Leetcode, etc..

# 

# 

# 2. Introduction

## **2.1 brief introduction**

Decoder platform is created for programmers.

User is able to add,edit and delete the questions which are added by himself only. Other users can only read these questions. Also, users can simply use a compiler built in our website and compile/run their code. Users can save multiple codes. One user can’t see the other user's code.

There are a lot of programming platforms out there each with their own contest, for one programmer it’s hard to keep a map of every contest date, time and duration. This project will help them to see all the contest information at a single place.

## **2.2 Tools / Technologies Used**

**Technologies :**

HTML5

CSS3

Bootstrap

ReactJS

Express with NodeJS

MongoDB

**Tool :**

Git

GitHub

Visual Studio Code

**Platforms:**

Local Development Server

# 

# 

# 3. Software Requirement Specifications

**Manage User Profile:**

R1.1 Login:

|  |  |
| --- | --- |
| Input | username(or email) and password |
| Output | Redirection according to success or failure |
| Description | User can log into system to get additional features  and save his/her data regarding to system |
| Process | Validates the given credentials |
| Precondition | User should have an account to login |

Registration:

|  |  |
| --- | --- |
| Input | Credentials of the user as asked |
| Output | Redirection according to success or failure |
| Description | User can log into system to get additional features  and save his/her data regarding to system |
| Process | Registers an account |
| Precondition | Credentials validation. If validation is successful then  account is registered into the system |

Forgot Password:

|  |  |
| --- | --- |
| Input | username(or email) and password |
| Output | Redirection according to success or failure |
| Description | User can log into system to get additional features  and save his/her data regarding to system |

Logout:

|  |  |
| --- | --- |
| Description | To logout from the system |
| Process | Session will be destroyed and The user will be  redirected to the home page |
| Precondition | Logged into the system |

Manage code:

|  |  |
| --- | --- |
| Input | Required credentials |
| Output | Confirmation message |
| Description | User can save, update, delete and share the code |
| Process | Save, update , delete , share code |
| Precondition | Authenticated |

Compile and Run the code:

|  |  |
| --- | --- |
| Input | Code you want to write |
| Output | Compile error or output |
| Description | User can compile and run the code |
| Process | Compile code in system |

Upcoming Contest:

|  |  |
| --- | --- |
| Input | Required button click(on various platform) |
| Output | Upcoming contest list based on filter |
| Description | Get some of the details like starting/ending date and  time of the contest from various sites |
| Process | Get contest list |
| Precondition | User should have an account |

Manage Question:

|  |  |
| --- | --- |
| Input | Required credentials(title,description,tag etc..) for save,update. Button click for Read, Delete. |
| Output | Confirmation message |
| Description | User can save, update, delete and read Question |
| Process | Save, update , delete , share Question |
| Precondition | Authenticated |

Show Question list:

|  |  |
| --- | --- |
| Input | Button Click |
| Output | List of all the questions |
| Description | Get list of all the question based on difficulty level and tags |
| Process | Searching the questions from database |
| Precondition | User should have an account |

# 

# 4. Design

## **4.1 Use Case Diagram**

# 

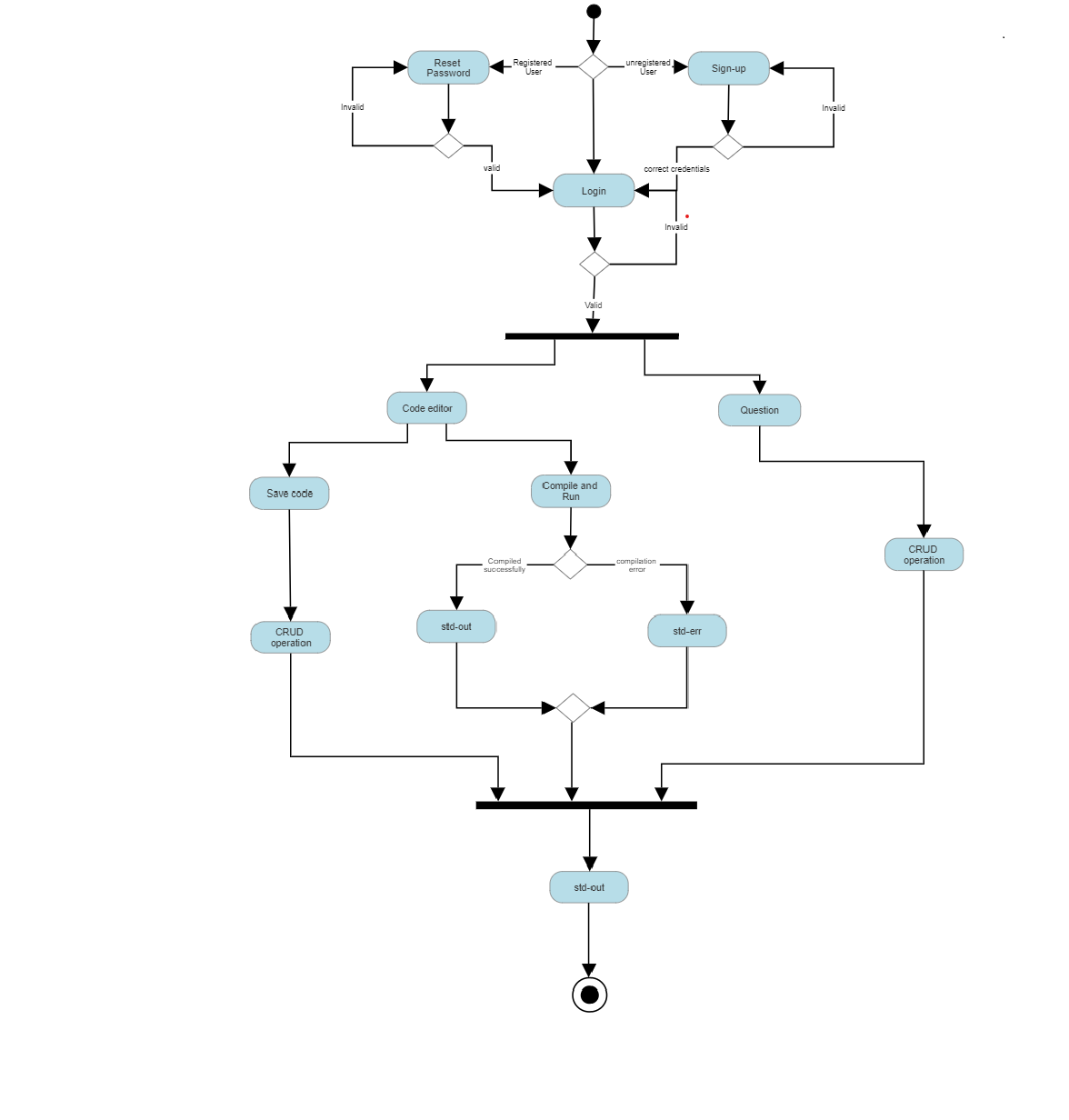
## **4.2 Class diagram**

# 

## 

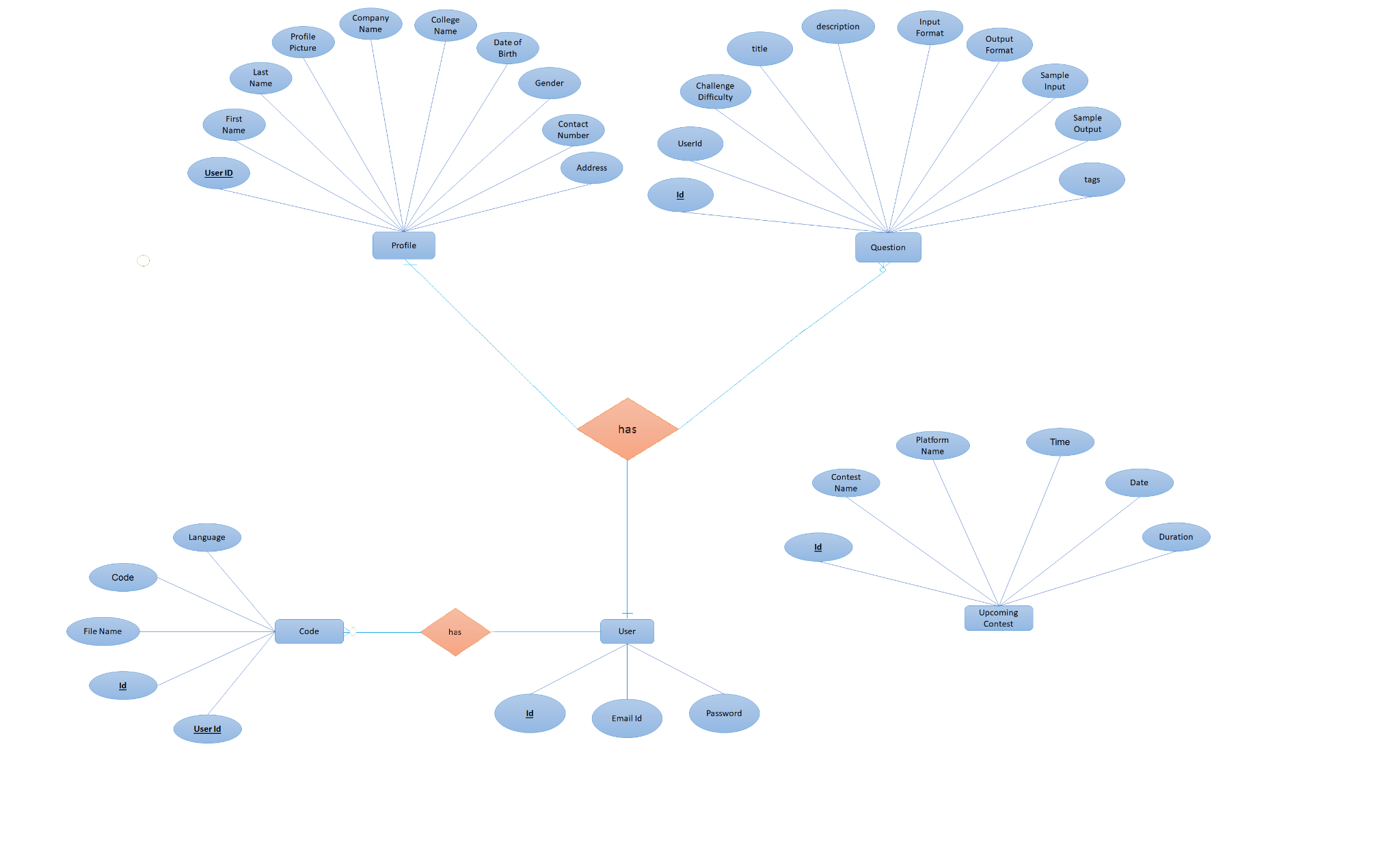
## 

## **4.3 Activity diagram**



## 

## **4.4 ER diagram**



## **4.5 Data dictionary**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User** | | | | | | |
| Sr. No. | Field Name | Data Type | Width | Required | Unique | PK/FK |
| 1 | Id | String | 100 | Yes | Yes | PK |
| 2 | Email | Email | 30 | Yes | Yes |  |
| 3 | Password | Password | 12 | Yes | No |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Profile | | | | | | | |
| **Sr. No.** | **Field Name** | **DataType** | **Width** | **Required** | **Unique** | **PK/FK** | **Referenced Table** |
| 1 | Id | String | 100 | Yes | Yes | PK |  |
| 2 | UserId | String | 100 | Yes | Yes | FK | User |
| 3 | First Name | String | 30 | Yes | No |  |  |
| 4 | Last Name | String | 30 | No | No |  |  |
| 5 | College Name | String | 100 | No | No |  |  |
| 6 | Date of Birth | Date | 30 | No | No |  |  |
| 7 | Image URL | String | 500 | No | No |  |  |
| 8 | gender | String | 10 | No | No |  |  |
| 9 | String | Address | 1000 | No | No |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Question** | | | | | | | |
| **Sr.No.** | **Field Name** | **Data Type** | **Width** | **Required** | **Unique** | **PK/FK** | **Referenced Table** |
| 1 | Id | String | 100 | Yes | Yes | PK |  |
| 2 | UserId | String | 100 | Yes | No | FK | User |
| 3 | Challenge Difficulty | String | 100 | Yes | No |  |  |
| 4 | Title | String | 100 | Yes | No |  |  |
| 5 | Description | String | 100 | Yes | No |  |  |
| 6 | Input Format | String | 100 | Yes | No |  |  |
| 7 | Output Format | String | 100 | Yes | No |  |  |
| 8 | Sample Input | String | 100 | Yes | No |  |  |
| 9 | Sample Output | String | 100 | Yes | No |  |  |
| 10 | Tags | String | 100 | Yes | No |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Code** | | | | | | | |
| **Sr.**  **No.** | **Field**  **Name** | **Data**  **Type** | **Width** | **Required** | **Unique** | **PK/FK** | **Referenced**  **Table** |
| 1 | Id | String | 100 | Yes | Yes | PK |  |
| 2 | UserId | String | 100 | Yes | No | FK | User |
| 3 | FileName | String | 100 | Yes | No |  |  |
| 4 | Code | String | 1000 | Yes | No |  |  |

# 

# **5. Implementation Detail**

1. **Sign-up / Login**

Sign-up functionality is successfully implemented. Users need to enter valid credentials details during the sign-up process. Proper validation messages are shown if a user gives incorrect/invalid credentials. Verification email(OTP) is sent to the user to avoid misuse of the email address hence unless the user doesn't enter OTP which is sent to the entered email address user wouldn't be able to register. Afterwards users will be able to login through either their unique email and password. If during login the user forgot the password then also one can update the password through OTP sent to the registered email address.

1. **Compiler API**

We've created a RESTful API in the backend which will help in

compiling the code and then running the code on a certain set of input and yielding the corresponding stdout/stderr.

1. **Question and Code**

Users can do CRUD(Create, Read, Update, Delete) operations on the questions. Users can also search problems by difficulty level(Easy, Medium, Hard) and by tags(dp,binary search, backtracking etc..). Users are able to compile, save and share the code.

1. **Upcoming Contests**

We've fetched future competitive programming contests and listed them down, users can sort them according to the website, search for particular contests, add contests to their google calendar using just one click.The data is fetched using free api provided by kontest.com.

# 

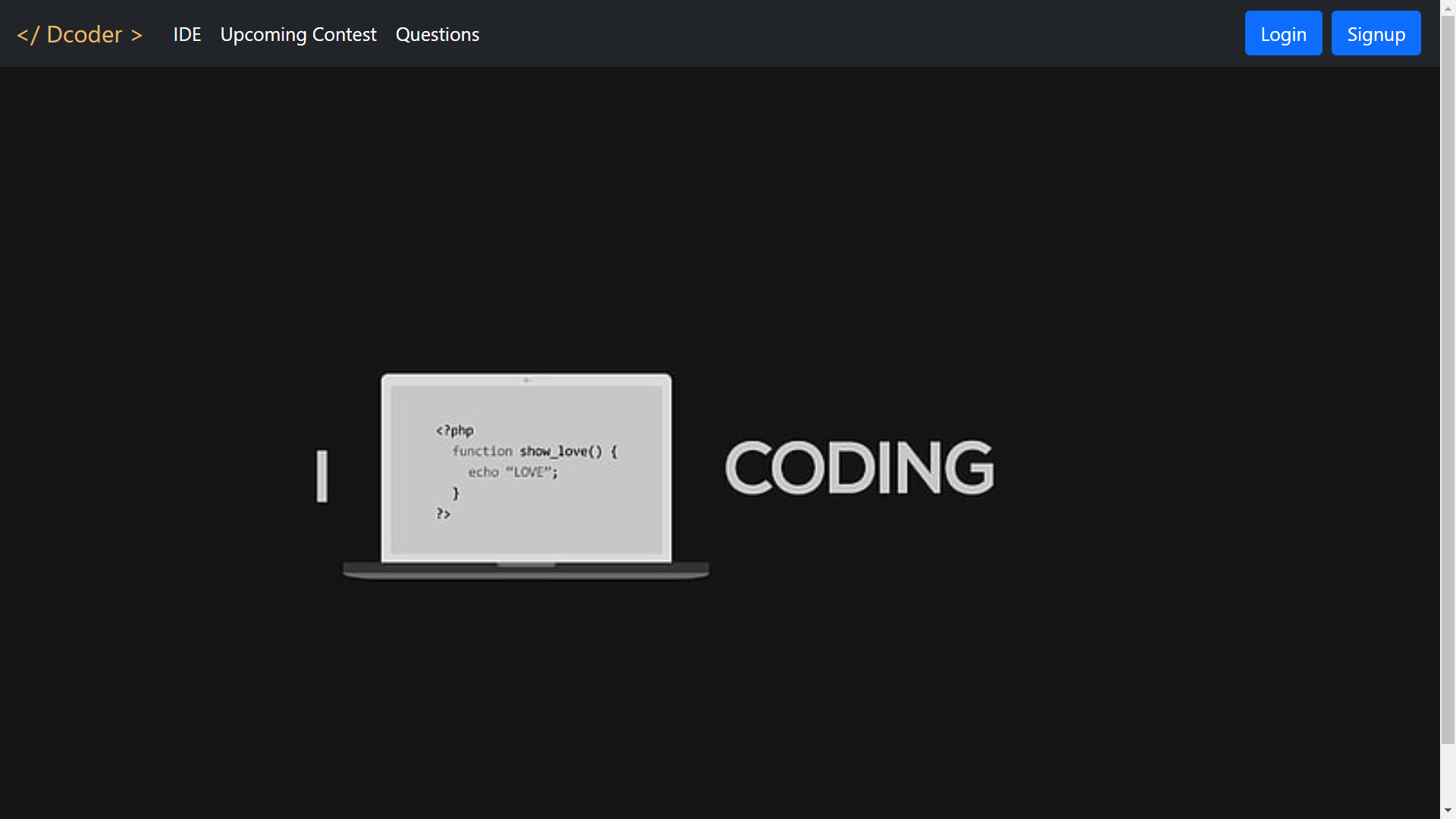
# 6. Testing

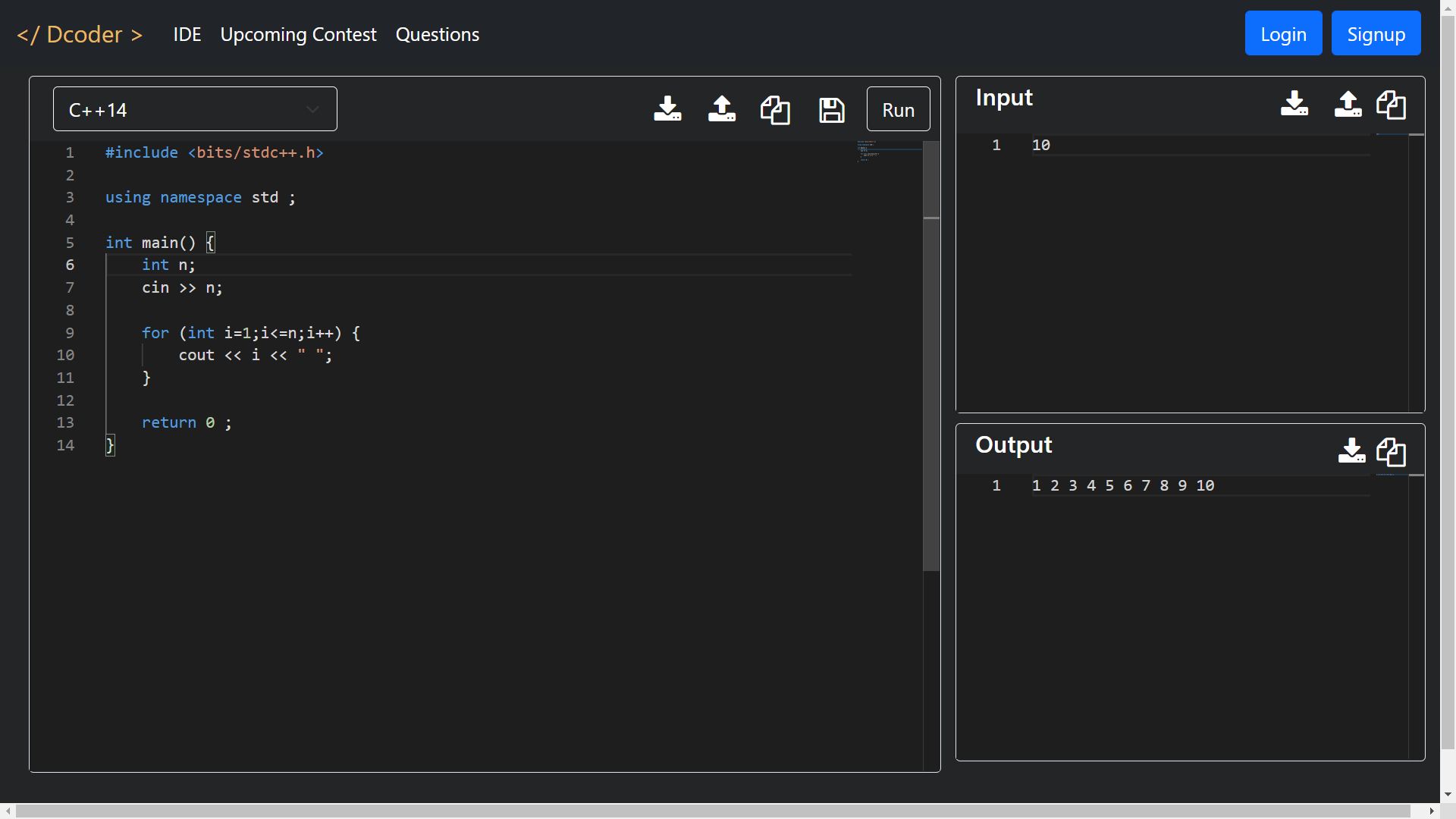
# 

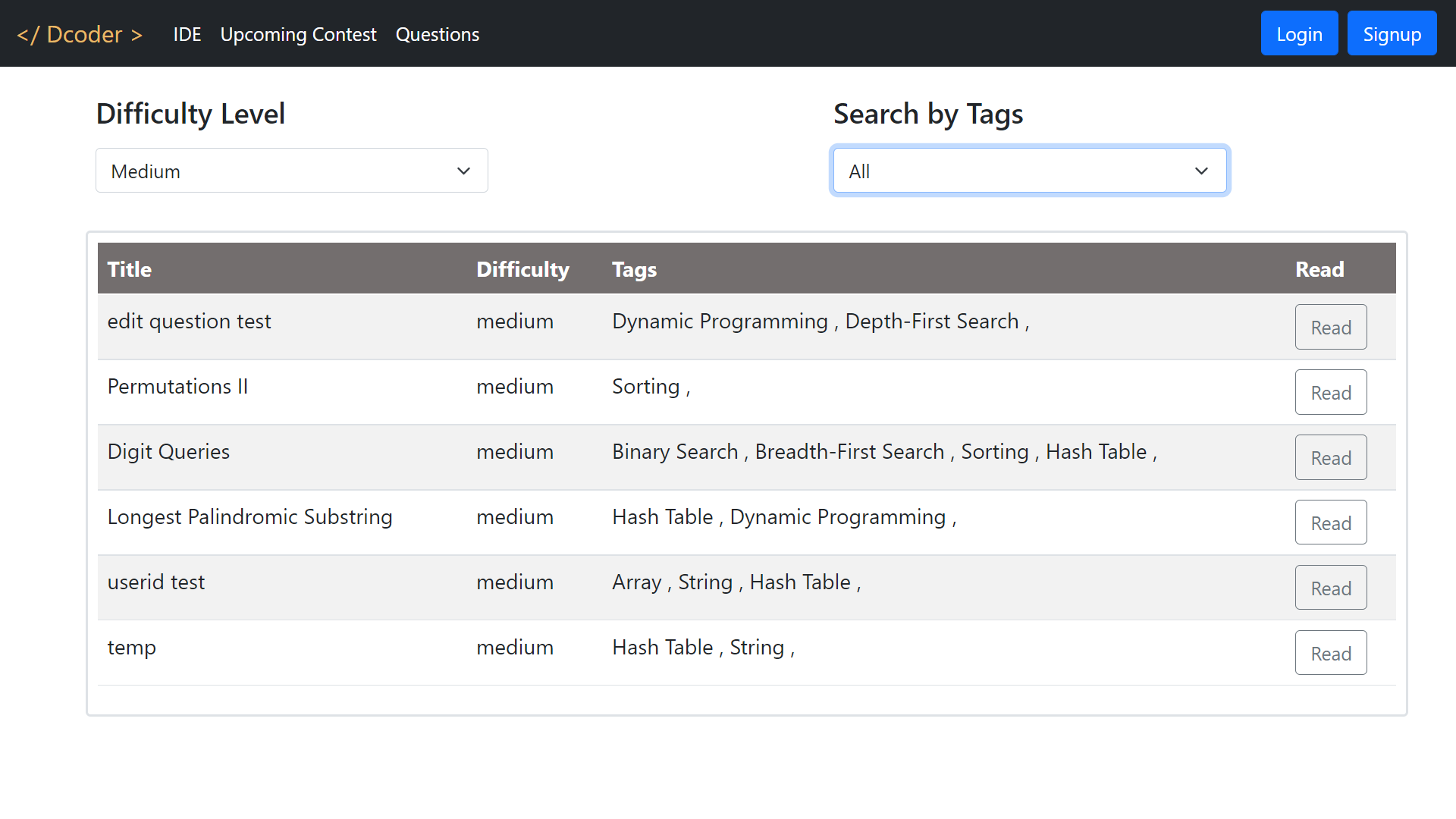
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Module** | **Field names** | **Expected output** | **Actual output** | **Status** |
| Login | Username &  Password (correct  credentials) | Redirect to Home  page | Redirect to Home  page | Success |
| Login | Username &  Password (incorrect  credentials) | Validation error  message and  stays on login  page | Validation error  message and  stays on login  page | Success |
| Sign-Up | Email (email which is already taken) | Warning message  and stay to sign up  page | Warning  message and  stays on signup  page | Success |
| Sign-Up | Correct credentials | Redirect to login page | Redirect to login page | Success |
| Code-editor | Trying to compile  code | Code compilation  successful and  corresponding  stderr/stdout is  given | As expected | Success |

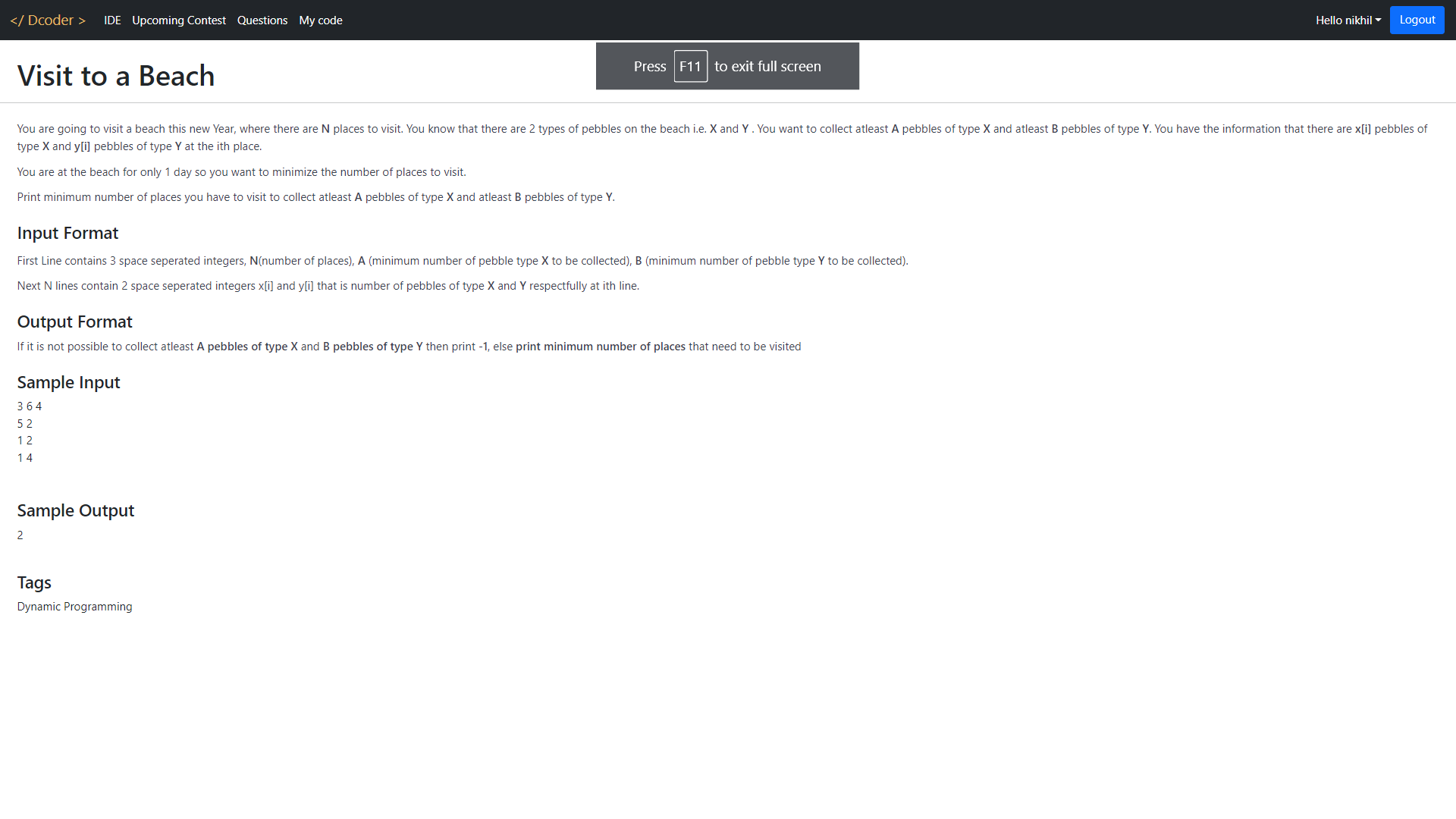
# 

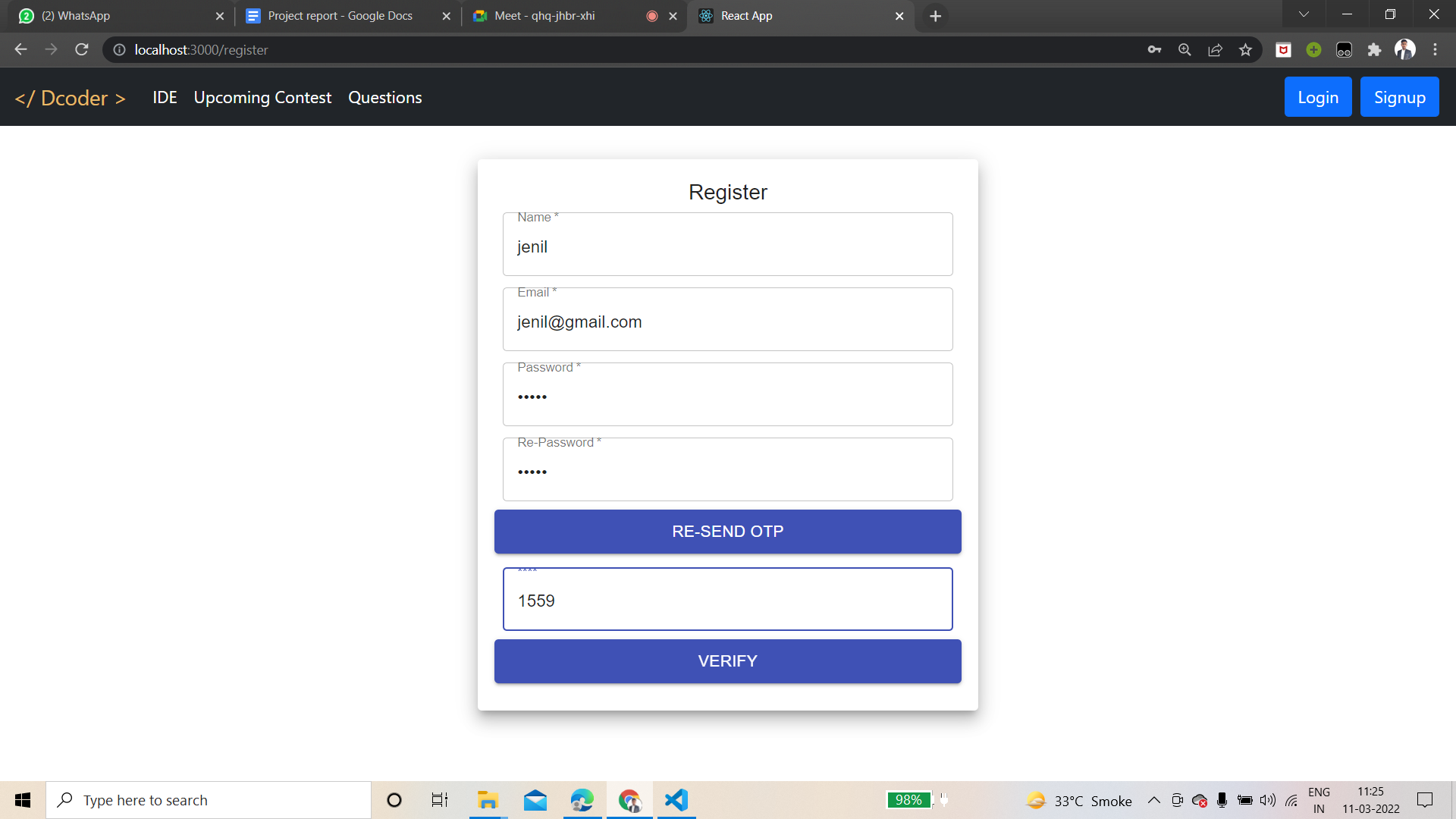
# **7. Screenshots**

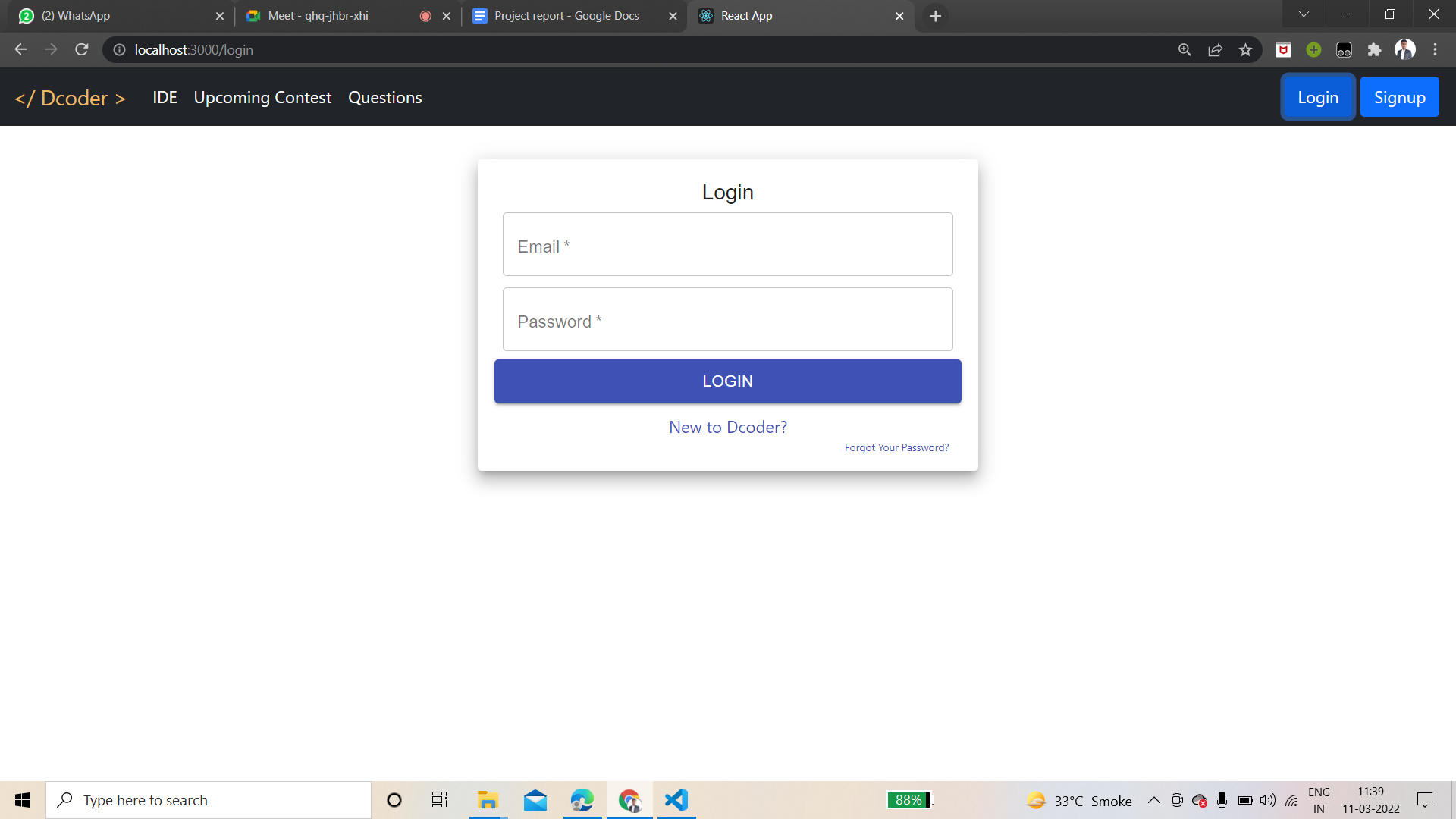
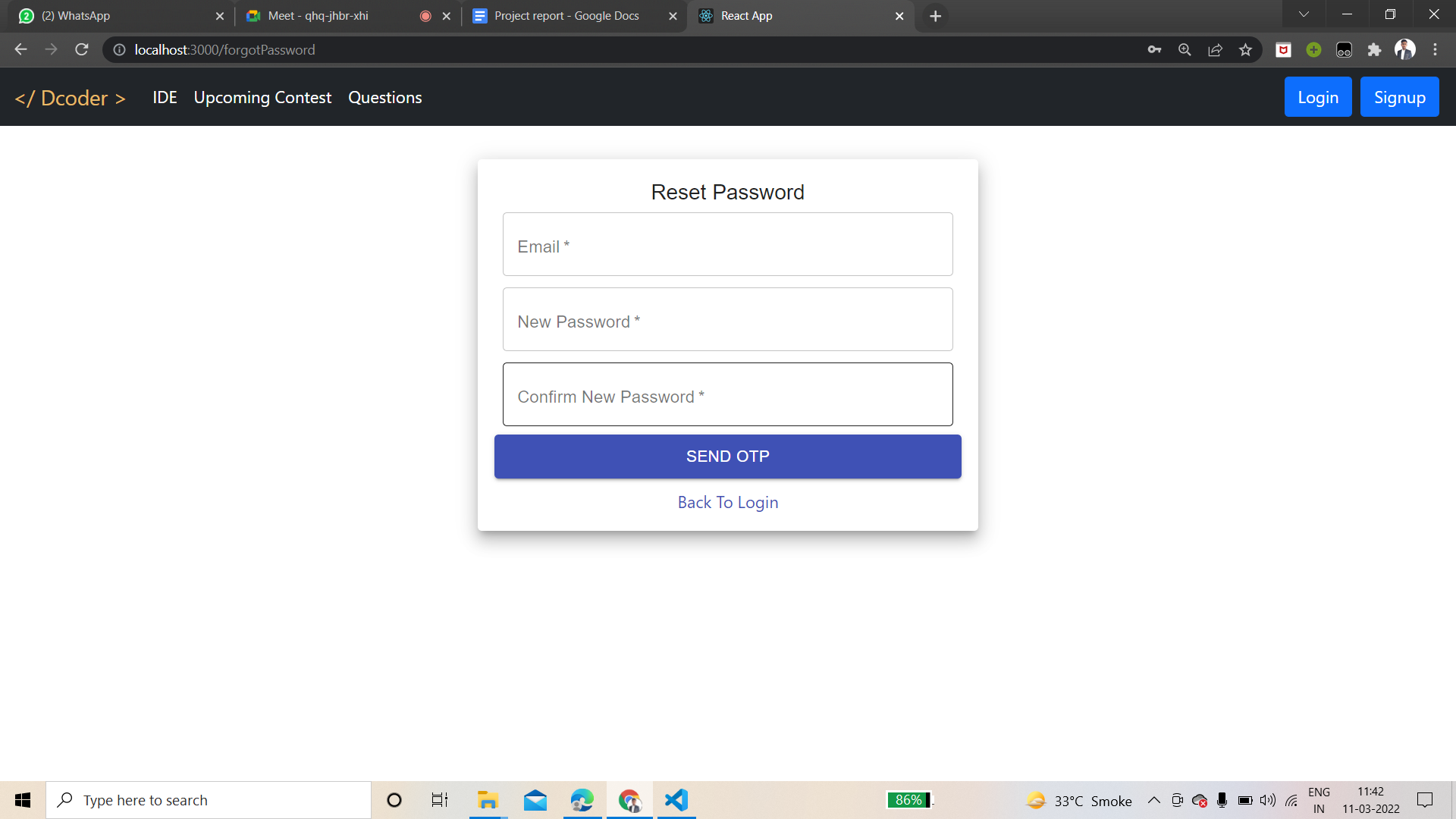


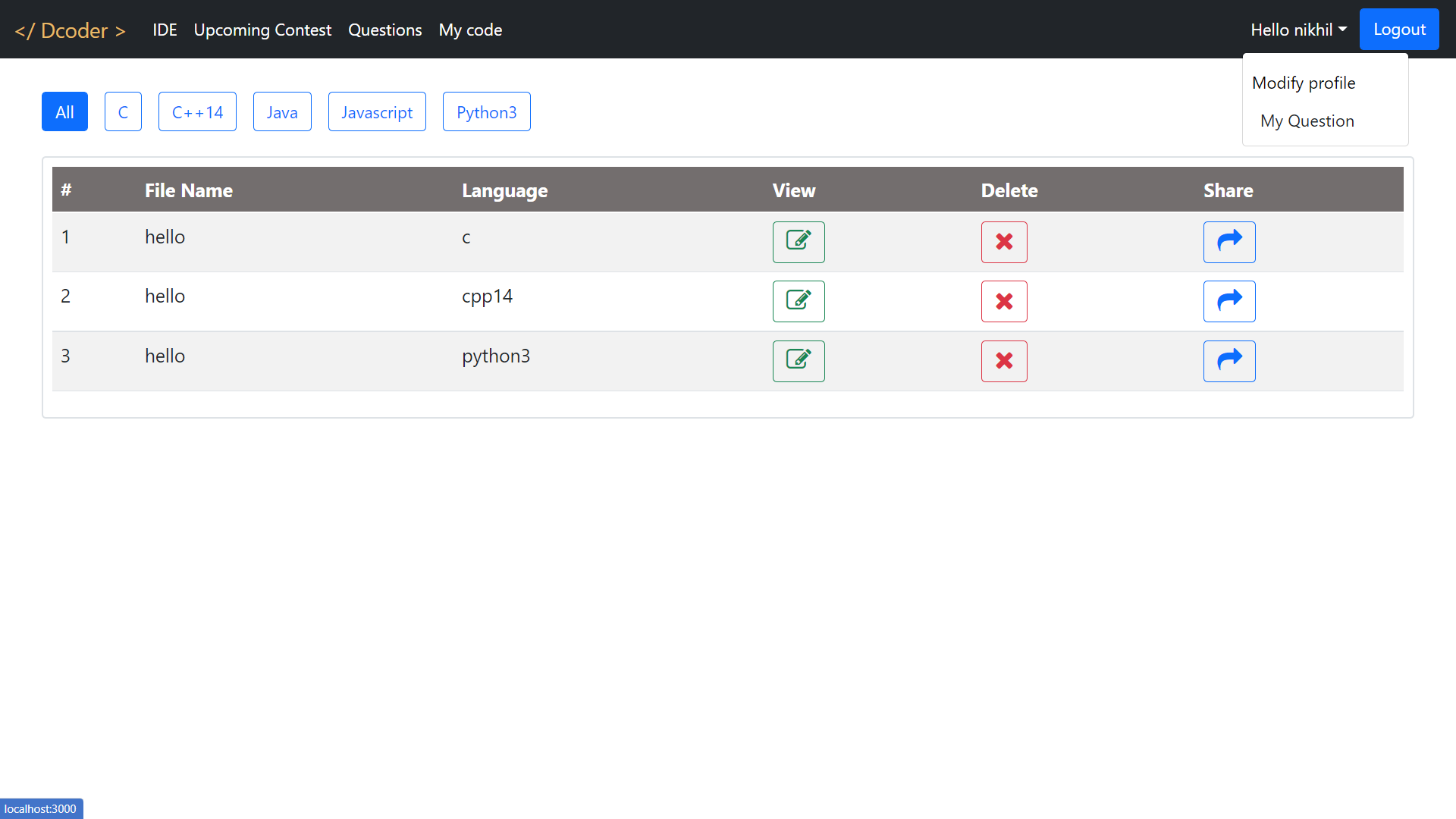


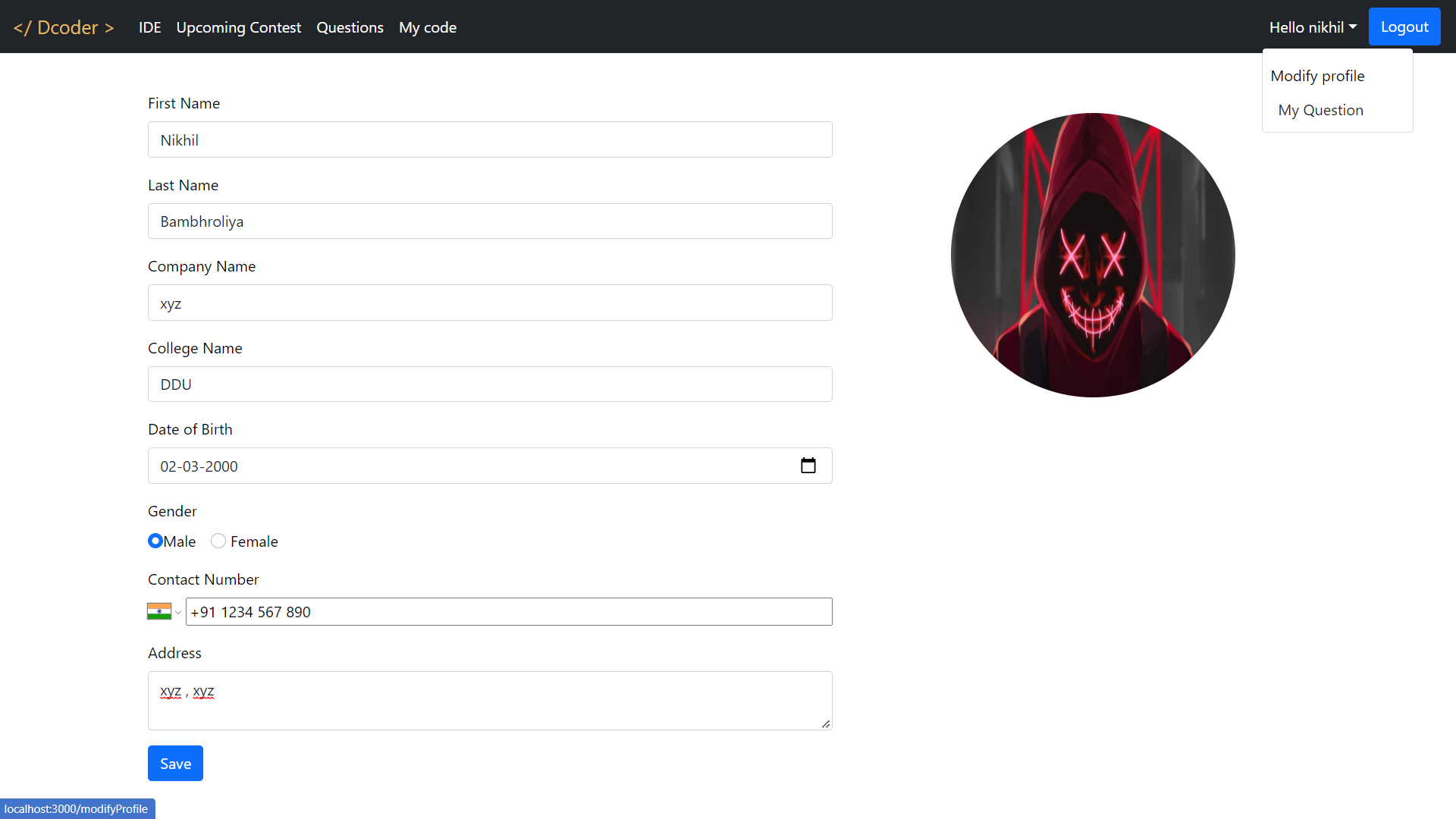


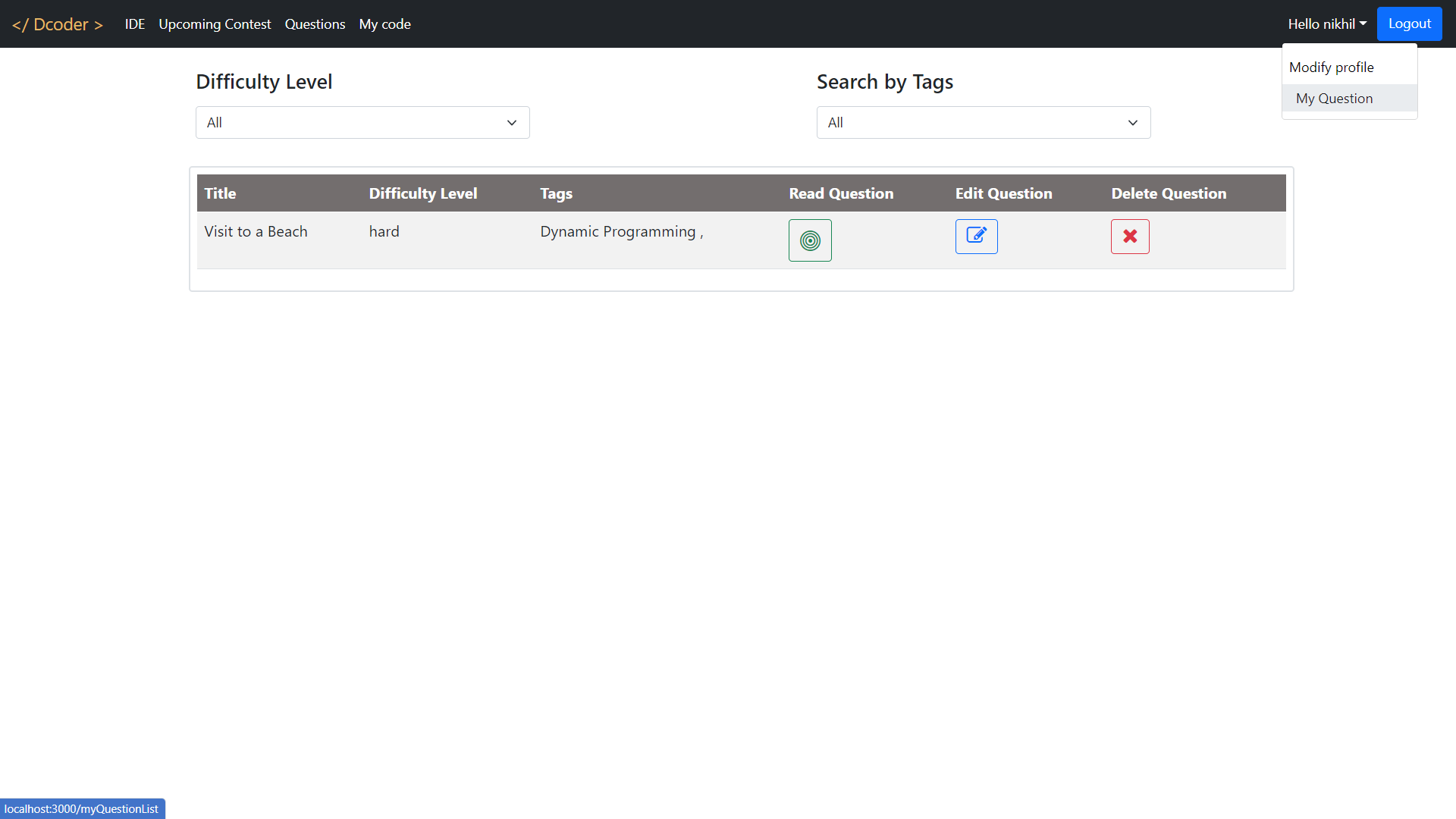


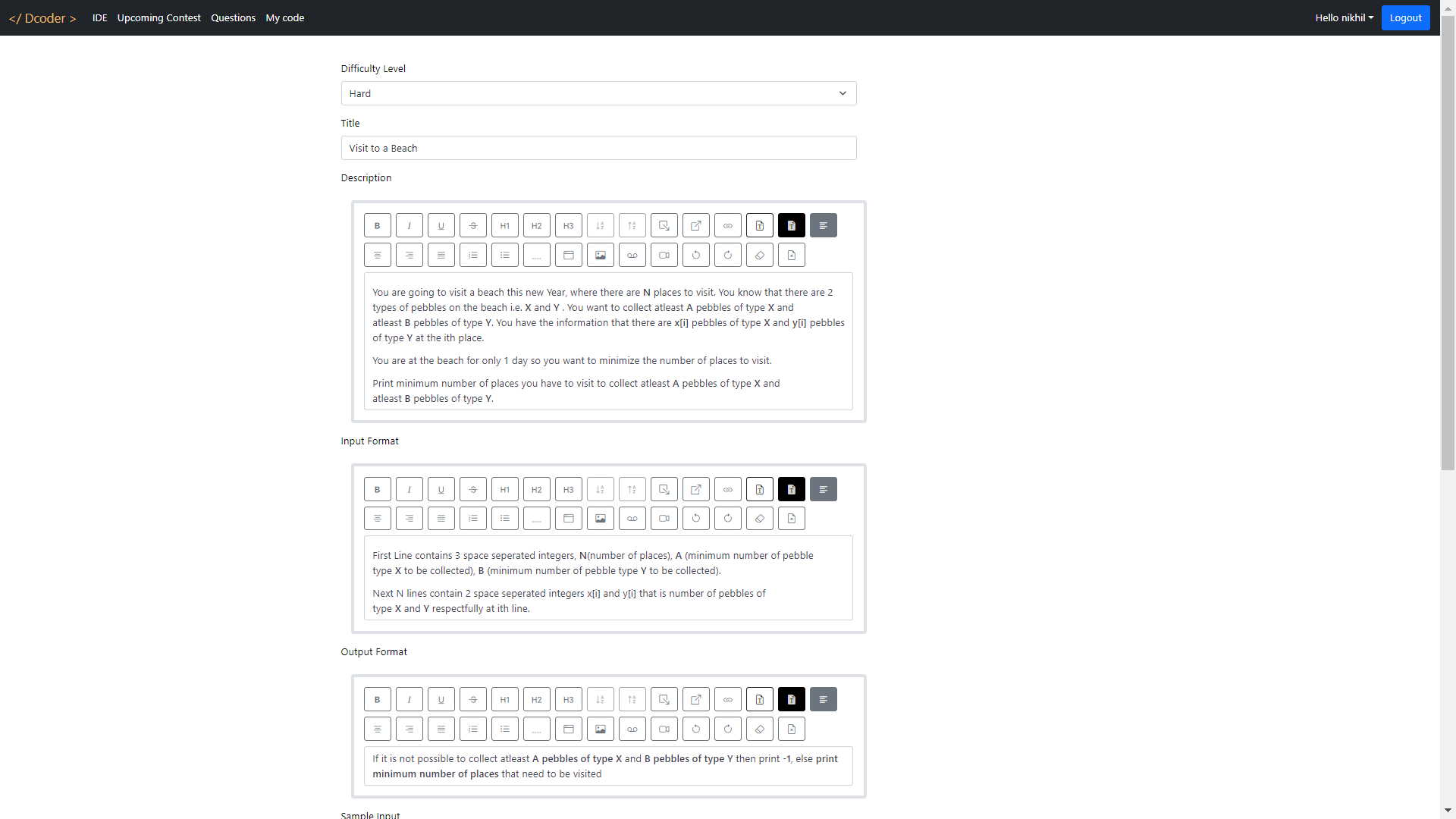










# 

# **8. Conclusion**

The solution is to provide an online tool that gives a compiler for

popular languages so that programmers don't have to waste their time in installing SDKs and compilers for different languages and setting up their own environment. Users can save their code at the server so no waste of storage space. Contest time table to get up to date with upcoming contest timings and duration.

# 

# **9. Limitation and Future Extension**

The future extension of this project should include support for more

programming languages, unlimited compilation requests, and free

compilation api for developers to create more and more competitive

programming helper tools. Also discussion form for doubts for user’s

question.

# **10. References**

- <https://getbootstrap.com/>

- <https://reactjs.org/>

- <https://ideone.com/>

- <https://www.jdoodle.com/>

- <https://www.kontests.net/>

-<https://www.npmjs.com/package/@monaco-editor/react>