Ministry of Higher Education
Modern Academy
Computer Science and Management
Technology in Maadi
Computer Science Department

CodeHex

Prepared by:

Abdelaziz Essam Abdelaziz
Ali Muhammed Ali
Emad Mahfouz Ahmed
Mazen Muhammed Abdelaziz
Abdullah Sayed Hanafi

Supervised By:

Academic Supervisor: Dr. Mohamed Awad

Assisting Supervisor: Eng. Mariam Medhat

Egypt (2022 - 2023)

Acknowledgement

We have taken a lot of effort into this project. However, the completion of this project could not have been possible without the participation and assistance of a lot of individuals contributing to this project.

Firstly, we would like to express our gratitude towards our university **"Modern Academy"** and its board members for their professionalism, supervision, and support that helped us grow on our educational journey.

"Prof Dr. Nabil Deabes" For the exceptional quality of education we have received that could not have been without his supervision which in return helped us prepare for our professional career.

"**Prof. Dr. Mahmoud Gadallah"** For his guidance and keenness on providing us with the best educational facilities along with the department staff that helped us so much throughout our academic years.

"**Dr. Mohamed Awad"** For her valuable guidance and support in completion of this project. And for always pushing us to do better. We are truly thankful to have been given the opportunity to learn from her and be inspired by her encouragement since day one.

Finally, special thanks to **"Eng. Mariam Medhat"** for her help and effort through the year and for always guiding us with patience and care.

All the help and support that we have received during the project implementation is very much appreciated and it is the reason we were finally able to present our idea and make it real, hoping it helps many people out there.

Abstract

CodeHex is a contest management system that enables users to create and manage contests, competitions, or tournaments. The system typically provides features such as registration, scheduling, score tracking, and communication tools to facilitate the smooth running of the event. Contest organizers can use the system to set rules, criteria for evaluation, and prizes, while participants can use it to submit entries, view results, and communicate with other contestants and organizers. A well-designed contest management system can help streamline the administration of a contest, reduce errors, improve transparency, and enhance engagement among participants. Contest management systems are used by a wide variety of organizations, including corporations, nonprofits, educational institutions, and government agencies. They are commonly employed to run contests for marketing purposes, such as to promote a new product or service, gather user-generated content, or boost engagement on social media platforms. Contests can also be used in the context of academic or scientific research, sports tournaments, and artistic or cultural events. One of the main benefits of using a contest management system is that it automates many of the tasks involved in organizing a contest, saving time and reducing the workload on organizers. For example, the system may enable participants to sign up online, submit their entries digitally, and receive automatic notifications about the contest schedule and results. The system can also provide real-time feedback to participants, enabling them to track their progress and adjust their strategies accordingly. Another benefit of using a contest management system is that it can improve the fairness and transparency of the contest. By providing a centralized platform for all participants to submit their entries and view the results, the system reduces the risk of bias or favoritism. It also allows organizers to enforce consistent rules and evaluation criteria across all

entries. Overall, a well-designed contest management system can enhance the experience of both participants and organizers, making it easier to run successful contests while minimizing the risk of errors or disputes. As technology continues to evolve, we can expect these systems to become even more sophisticated and user-friendly, further expanding the possibilities for running innovative and engaging contest.

Structure of the document:

This document will be split into Six main chapters.

Chapter 1: Introduction.

Chapter 2: Survey and Historical data.

Chapter 3: Software life cycle and Analysis.

Chapter 4: Tools.

Chapter 5: System Implementation .

Chapter 6: Conclusion and future work.

Table of contents:

Chapter four (Tools):

Chapter five (System implementation):

Chapter six (Conclusion & Future work):

List of figures:

Chapter 1
Introduction

1.1 Overview :-

CodeHex is an easy-to-use contest management system that helps organizers create and manage programming and problem-solving competitions. It offers a range of features such as registration, scheduling, score tracking, and communication tools that make event management easier and more efficient. CodeHex is customizable and automates time-consuming tasks, saving organizers time and reducing the risk of errors. It also promotes collaboration among participants and enhances transparency and fairness in events. When we began our final year graduation project, we wanted to create something that would solve a real-world problem in a field that we love. The result was CodeHex, which not only benefits organizers but also students who participate in programming and problem-solving competitions. Students can sign up for events online, submit entries digitally, and receive automatic notifications about the competition schedule and results. Additionally, CodeHex provides real-time feedback, enabling students to track their progress and learn from other participants. Overall, CodeHex is a valuable resource that streamlines the competition experience, fosters growth and development in programming communities, and solves a pressing issue facing event organizers.

1.2 Problem Definition :-

1-Traditional methods of organizing contests involve numerous manual tasks that are time-consuming, error-prone, and challenging to manage.

2-Many contests suffer from issues such as bias, favoritism, and inconsistent rules and evaluation criteria, which can lead to dissatisfaction among participants and hinder the success of the contest.

3-The lack of a streamlined and efficient platform for managing contests makes it difficult to run successful contests with high participation and engagement levels.

4-Organizers have to spend significant time and resources on managing the various aspects of a contest, which can be a significant workload.

5-Participants face challenges in keeping track of their progress and communicating with other participants and organizers during a contest.

6-The lack of transparency and fairness in many contests can lead to disputes and negatively impact the reputation of the organizer.

7-There is a need for a more efficient and effective way of managing contests that can benefit various organizations across different domains that hold contests for different purposes.

1.3 Solution Approach :-

- It will be easy for organizers to create and manage competitive programming contests.

- The student will find a place to improve his knowledge and performance in programing

- Automate many of the tasks involved in organizing a contest, saving time and reducing the workload on organizers. For example, the system may enable participants to sign up online, submit their entries digitally, and receive automatic notifications about the contest schedule and results.

- Encrypt user data when stored in the database to prevent unauthorized access.

- Enhance the experience of both participants and organizers, making it easier to run successful contests while minimizing the risk of errors or disputes.

- Implement a database to store all necessary information, such as user details, contest details, and submissions, in order to retrieve data quickly and efficiently.

1.4 Objectives of the project :-

1- To design and develop a user-friendly contest management system that enables organizers to create and manage contests and participants to register and submit their entries.

2- To incorporate features such as scheduling, score tracking, and communication tools to facilitate the smooth running of the event and enhance engagement among participants.

3- To ensure that the contest management system is secure, reliable, and scalable to accommodate a large number of participants and entries.

4- To improve the fairness and transparency of the contest by providing a centralized platform for all participants to submit their entries and view the results, reducing the risk of bias or favoritism.

5- To automate many of the tasks involved in organizing a contest, saving time and reducing the workload on organizers, enabling them to focus on other aspects of the event.

6- To provide real-time feedback to participants, enabling them to track their progress and adjust their strategies accordingly.

7- To enable organizers to enforce consistent rules and evaluation criteria across all entries.

8- To create a system that can be used by a wide variety of organizations, including corporations, nonprofits, educational institutions, and government agencies.

1.5 Advantages:

1- Streamlined Administration: CodeHex automates many of the tasks involved in organizing a contest, saving time and reducing the workload on organizers. For example, participants can sign up online, submit entries digitally, and receive automatic notifications about the contest schedule and results.

2- Enhanced Engagement: CodeHex provides a centralized platform for participants to communicate with other contestants and organizers, enhancing engagement and fostering a sense of community.

3- Improved Transparency: By providing a centralized platform for all participants to submit their entries and view the results, CodeHex reduces the risk of bias or favoritism and allows organizers to enforce consistent rules and evaluation criteria across all entries, improving transparency.

4- Scalability: CodeHex is highly scalable and can handle large volumes of contestants and submissions simultaneously, making it suitable for contests of any size.

5- Customizability: CodeHex offers customizable features and functionalities tailored to specific competition needs, enabling organizers to set rules, criteria for evaluation, and prizes according to their unique requirements.

6- Security: CodeHex ensures secure data storage and prevents hacking attempts during the contest, ensuring the confidential information of both organizers and participants is protected.

Chapter 2

Survey &

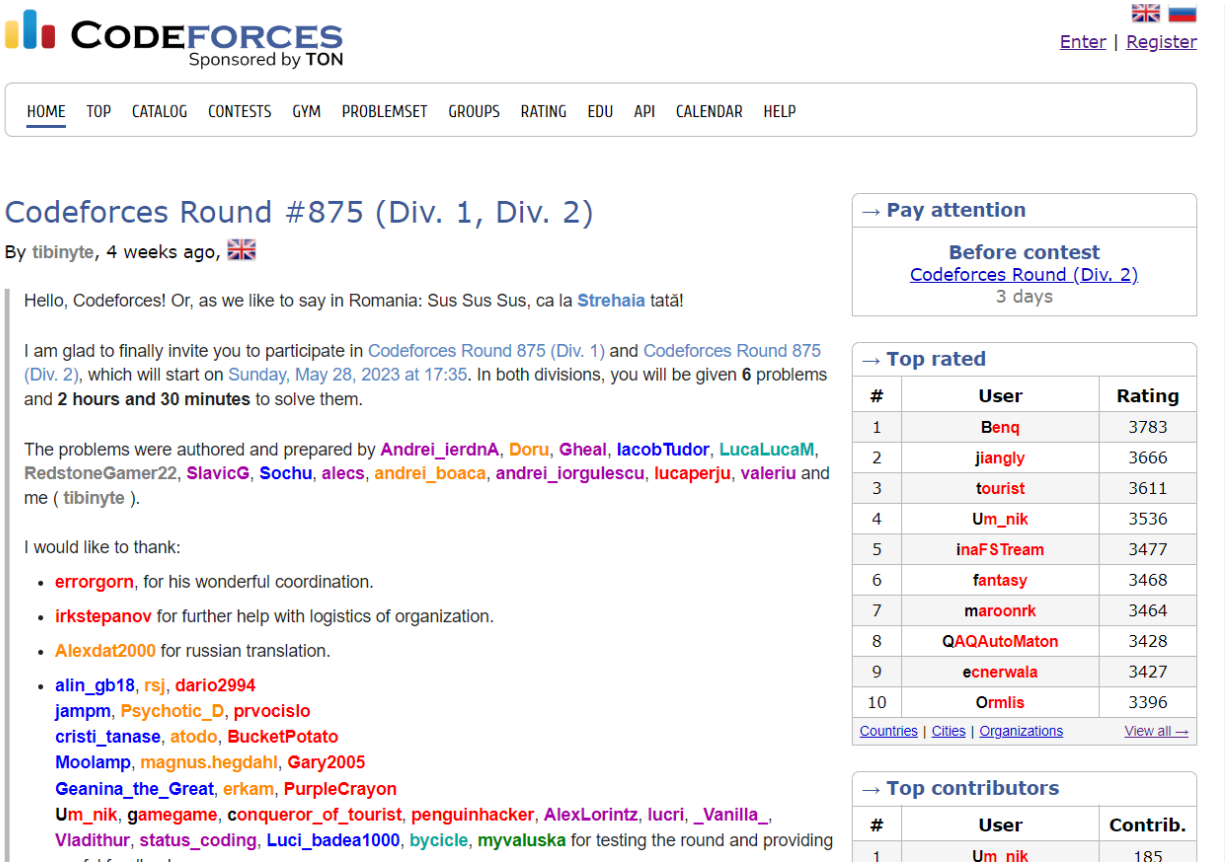
Historical data

2.1 Historical data:

We take some ideas from these websites, we adjust and add more features to be more easier and flexible.

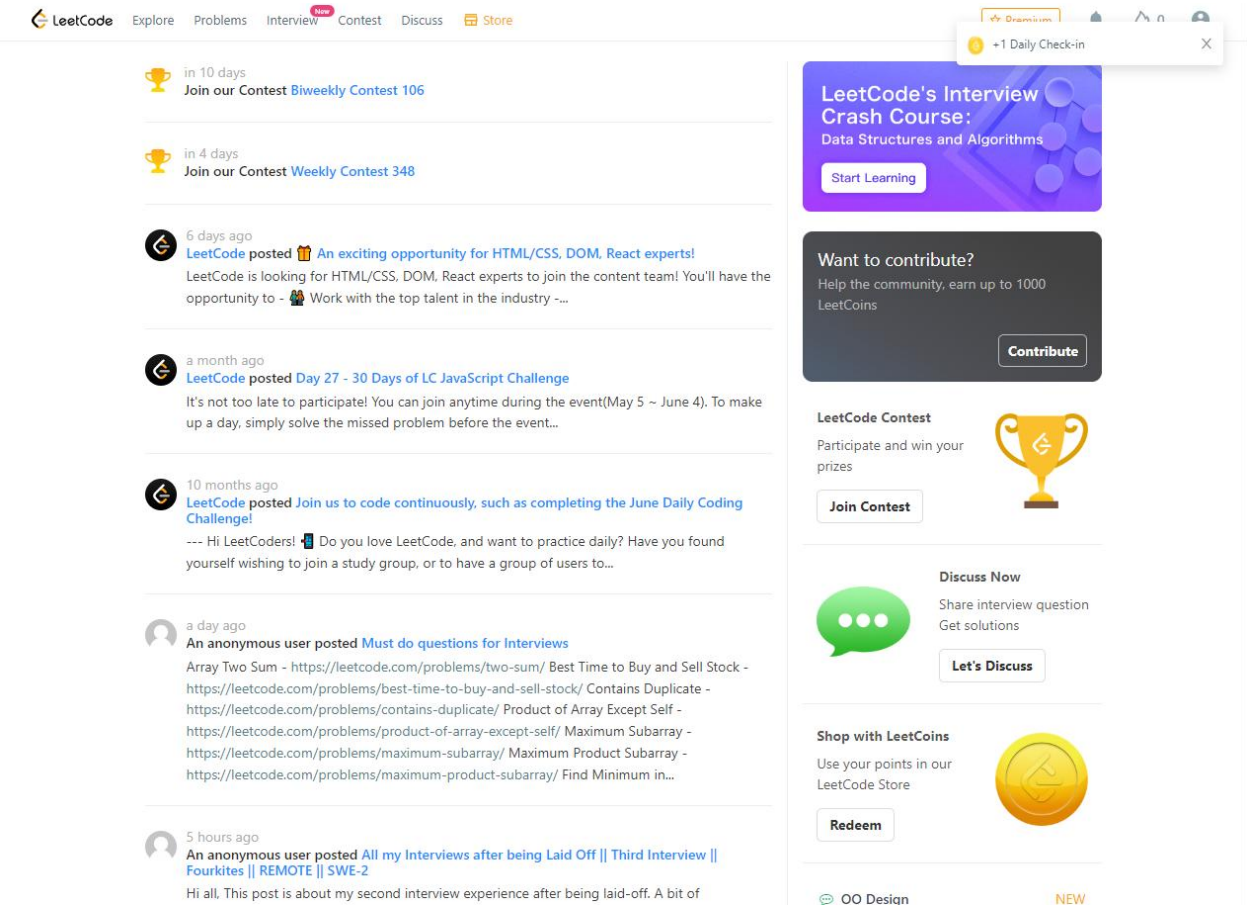
2.1.1 Codeforces :

Codeforces is an online platform that hosts competitive programming contests for individuals and teams. It offers a variety of algorithmic problems of varying difficulty levels that test participants' problem-solving skills and coding ability. Codeforces also provides a community forum for discussion, editorial solutions to past contests, and user ratings and rankings based on their performance in the contests. The platform is widely used by programmers, computer science students, and professionals around the world to test and improve their coding skills.



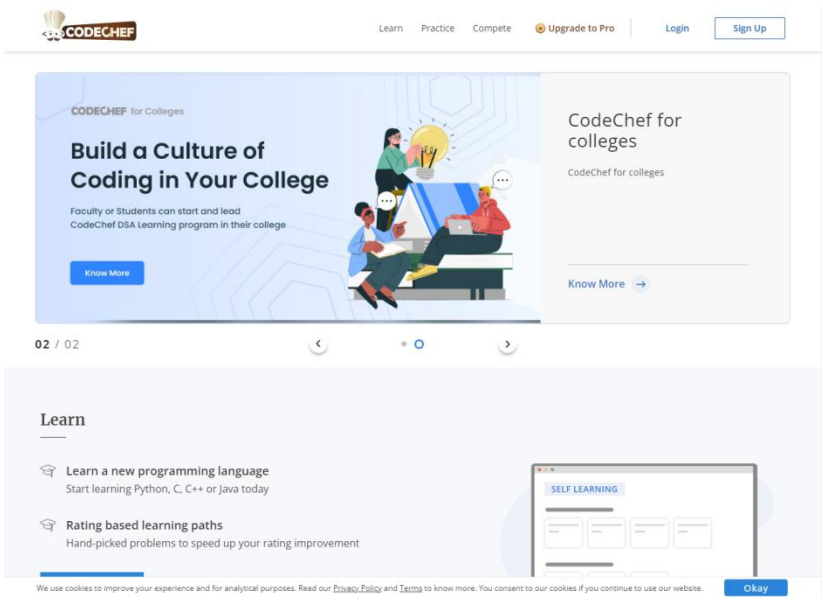
2.1.2 Leetcode:

LeetCode is an online platform that provides a collection of coding challenges, interview questions, and other coding-related resources to help individuals improve their coding skills and prepare for technical interviews. LeetCode's challenges cover a wide range of topics and difficulty levels, and the platform is frequently used by programmers, software engineers, and computer science students to practice coding, learn new algorithms and data structures, and demonstrate their coding proficiency to potential employers.

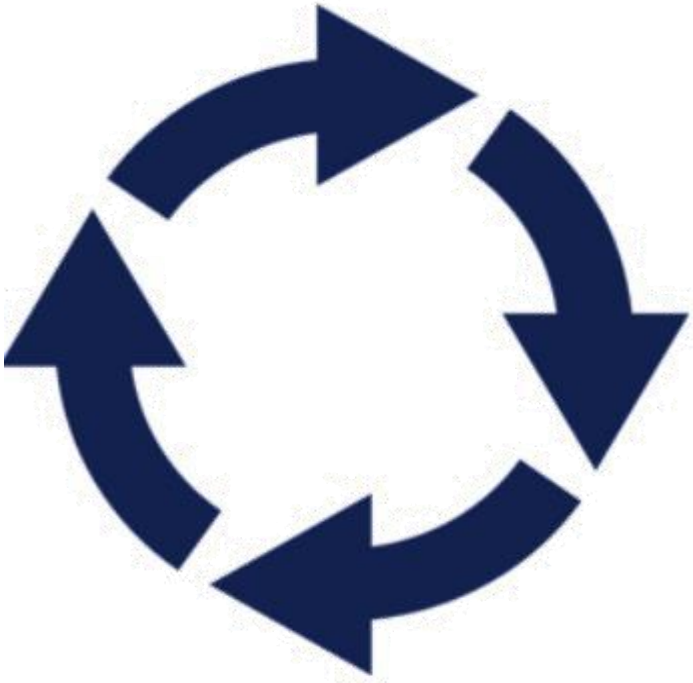


2.1.3 Codechef:

CodeChef is an online competitive programming platform that hosts monthly coding contests and challenges for programmers of all skill levels. The platform also offers a vast collection of practice problems and tutorials that cover various programming concepts, algorithms, and data structures. CodeChef's contests and challenges are designed to improve the coding skills of its users and help them prepare for coding competitions and technical interviews. The platform is popular among programmers, computer science students, and software engineers who want to enhance their problem-solving abilities and coding proficiency.



Chapter 3
Software life cycle & Analysis



3.1 Project Development Methodology (phases) :

The main characteristics of the Waterfall model are a sequential progression through the different stages of a project from initiation to the delivery phase. The waterfall model does have its limitations and because of this, there have been many spin off models created over the years.

The Waterfall model consists of the following phases:

- Introduction
- Analysis
- Design
- Coding
- Testing
- Implementation And Maintenance

The waterfall model used with the SDLC progresses through the six phases. You can picture this process as a waterfall with one phase flowing into the next.

- **Planning Phase** The purpose of the initiation phase is to conduct an initial high-level investigation of the business need and come up with a recommendation for the solution. Once approved by the management team, stakeholders, client or project sponsor, it will proceed to the next phase.

- **Analysis Phase** The purpose of the requirements analysis phase is to conduct a detailed analysis of the current business needs and identify what options are available to achieve those business needs. During the Analysis Phase, the Business Analyst will create the Business Requirements Document (BRD).

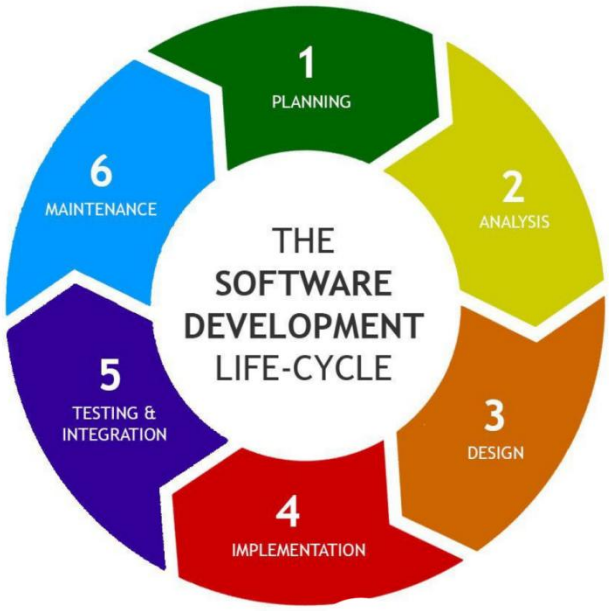
- **Design Phase** the purpose of the design phase is to identify and document a solution that will be constructed including technical and procedural specifications. A design document will be created that should include but not limited to technical, environmental, data, program, procedural, testing specifications.

- **Coding Phase** The construction or development phase is where a resource will take the design document created during the design phase and translate it into a functional program or system.

- **Testing Phase** The purpose of the testing phase is to test the system and related procedures that meet the requirements specified by the stakeholders and documented in the BRD, design plan, and testing plan.

- **Implementation and Maintenance Phase** the purpose of the implementation phase is to release a fully tested and operational product to an end user or customer. The product should meet all the requirements that were documented in the BRD and pass the testing phase before it can be released to a production environment.

The systems development life cycle (SDLC), or software development process in systems engineering, information systems and software engineering, is a process of creating or altering information systems, and the models and methodologies that people use to develop these systems. It consists of a set of steps or phases in which each phase of the SDLC uses the results of the previous one.



3.2 Planning:

Code Hex is a Desktop application and web based that In the context of competitive programming, CodeHex refers to a software tool called "Programming Contest Control Center". This tool is used to manage and automate the process of conducting programming contests, including tasks such as distributing problems, collecting solutions, and managing scoring. It provides a web-based interface for participants to submit their programs and check their results, and also allows administrators to monitor the contest progress and make adjustments as needed. Overall, CodeHex is a powerful and widely-used tool for organizing and executing programming competitions of various types and sizes.

3.3 Analysis:

3.3.1 Requirements Analysis:

Extracting the requirements of a desired software product is the first task in creating it. While customers probably believe they know what the software is to do, it may require skill and experience in software engineering to recognize incomplete, ambiguous or contradictory requirements.

3.3.2 System Requirements:

• Hardware

To make the website available on the internet and the Desktop application to be used by the instructors and students we need a high quality server that works on the internet.

•Software

1-For the Pc need them with Windows 7 or Higher for the web application but we request higher for the Desktop application to have higher performance.

2-For constructing the database we need MS Server.

3- For designing the interface of the web application we use HTML, CSS, C# and .Net Core

3.3.3 Software Requirements Specification :

- Functional Requirements : -

System Admin :

- Login
- Add Contest
- Edit Contest
- Delete contest
- Add Problem
- Edit Problem
- Delete Problem
- Ban Cheaters

System User :

- Login
- Register in Contest
- View Contest
- Submit Solution
- Get Submission State

- Nonfunctional Requirements:-

- **Usability:**

The system have friendly interface that make the user doesn't need any training.

- **Availability:**

The system will be available on 24 hours on 7 days this will be on the semester only.

- **Supportability:**

The system will be built using many tools like MS Server for database, HTML and C# for website & Desktop Application.

3.3.4 System Users:

In the system we have three main users:

1- Admin.

2- User.

3.3.5 Tools:

- **For Web design (Front-end) using Visual Studio Code:**

1- HTML Language (Hyper Text Markup Language).
2- CSS (Styling Language).
3- Java Script (Scripting Language).
4- JQuery (java Script Library).
5- Bootstrap (Front-end Frame work).

- **For Web Development (Back-end) using Visual Studio :**

1- C# (programming language).

- **For Desktop Application using Android Studio:**

1- C# (programming language).

- **For databases using SSMS:**

1- MS Server (for storing the social network data).

3.4 System Design :

The purpose of Design phase is to plan a solution for a problem specified by the requirements. System design aims to identify the modules that should be exist in the system, the specification of those modules and how they interact with each other to produce the results. The goal of the design process is to produce a model that can be used later to build that system. The produced model is called design of the system.

System design is the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements. Normally, the design proceeds in two stages:

• Physical design is a graphical representation of a system showing the system's internal and external entities and the flow of data into and out of these entities. An internal entity is an entity within the system that transforms data.

• Database design: It's the implementation of the schema into a database.

To represent the physical design of the system, we use diagrams like use case diagrams, etc.

3.4.1 Entity Relationship Diagram (ERD).

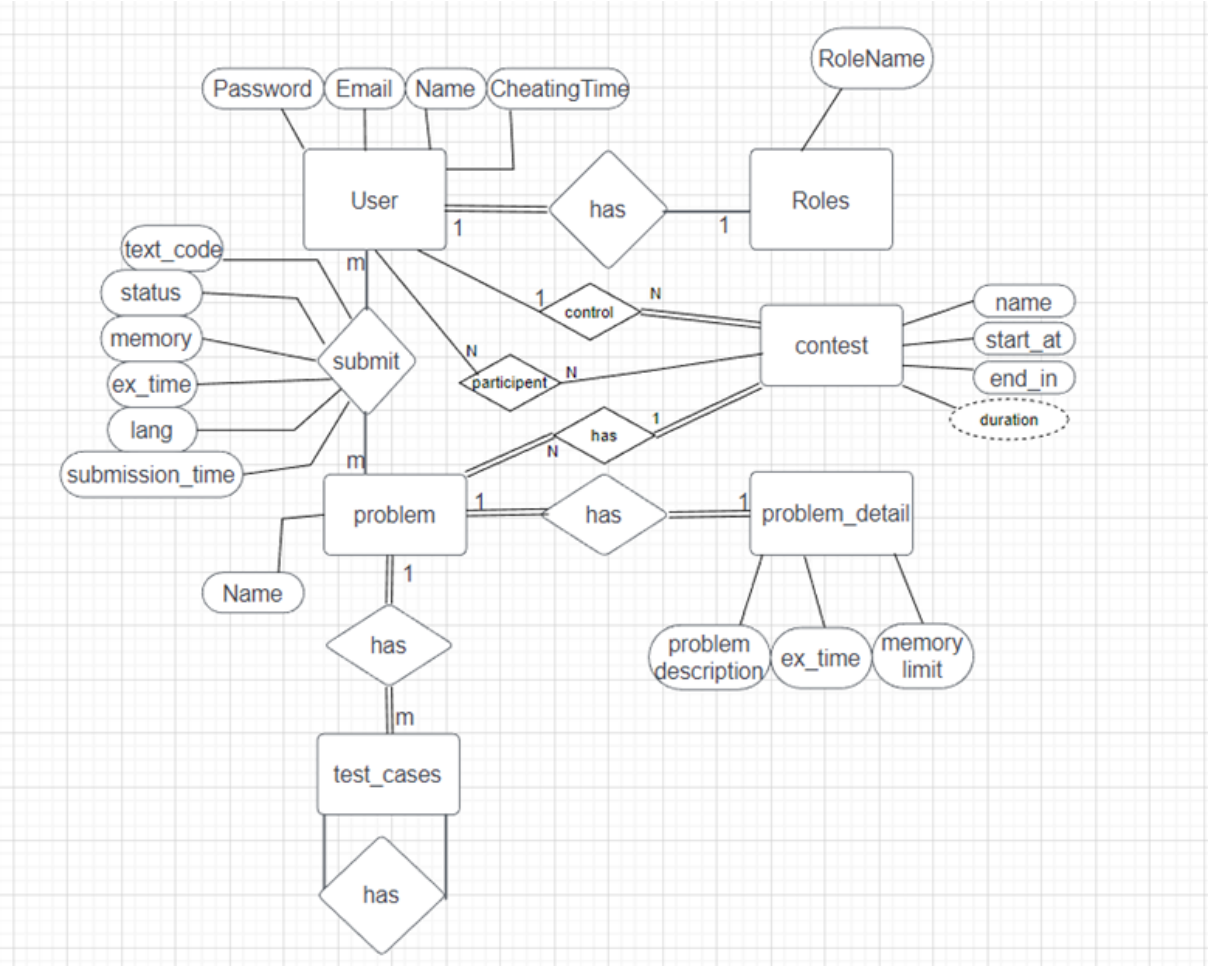


Figure 3.1 ERD diagram

3.4.2 The relation between tables in our Database (Schema).

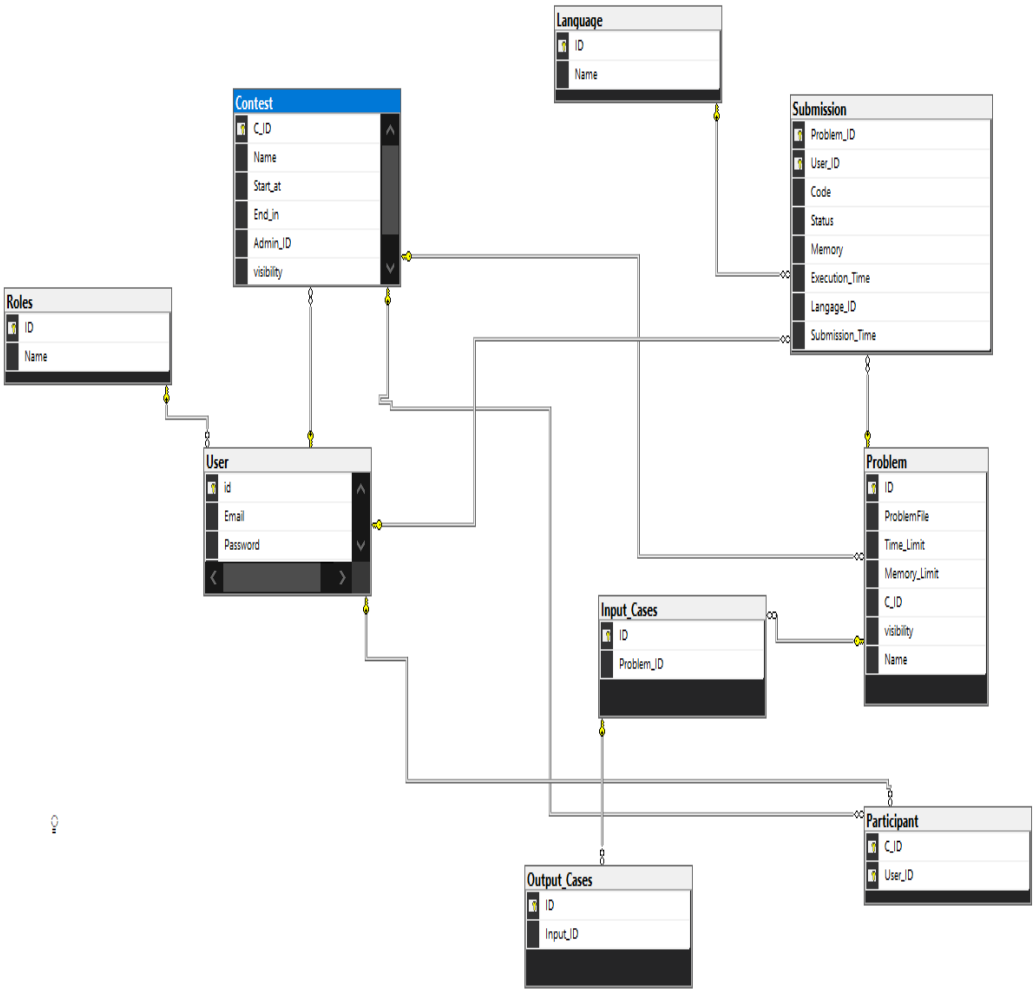


Figure 3.2 Schema of database

3.4.3 Tables Schema.

User table:

Column Name	Data Type	Allow Nulls
id	int	☐
Email	varchar(64)	☐
Password	varchar(64)	☐
R_ID	int	☐
UserName	varchar(64)	☐
Name	varchar(64)	☐
IsValid	tinyint	☐
CheatTimes	tinyint	☐

Figure 3.3 User table

Submission table:

Column Name	Data Type	Allow Nulls
Problem_ID	int	☐
User_ID	int	☐
Code	varchar(MAX)	☐
Status	tinyint	☐
Memory	int	☐
Execution_Time	float	☐
Langage_ID	int	☐
Submission_Time	smalldatetime	☐
		☐

Figure 3.4 Submission table

Roles table:

	Column Name	Data Type	Allow Nulls
🔑	ID	int	☐
	Name	varchar(30)	☐
			☐

Figure 3.5 Roles table

Problem table:

	Column Name	Data Type	Allow Nulls
🔑	ID	int	☐
	ProblemFile	varchar(MAX)	☐
	Time_Limit	decimal(18, 0)	☐
	Memory_Limit	int	☐
	C_ID	int	☐
	visibility	bit	☐
	Name	varchar(100)	☑
			☐

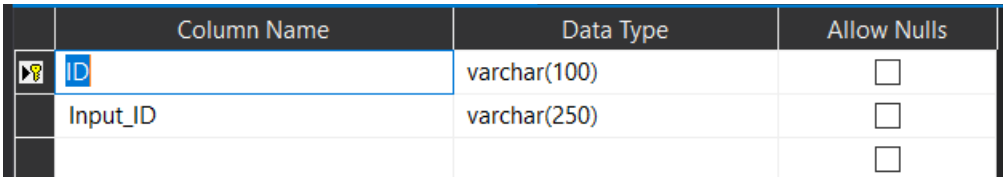
Figure 3.6 Problem table

Participant table:

	Column Name	Data Type	Allow Nulls
🔑	C_ID	int	☐
🔑	User_ID	int	☐
			☐

Figure 3.7 Participant table

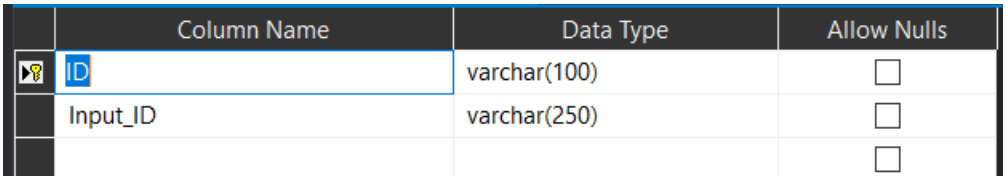
Output_Cases table:



Column Name	Data Type	Allow Nulls
ID	varchar(100)	☐
Input_ID	varchar(250)	☐
		☐

Figure 3.8 Output_Cases table

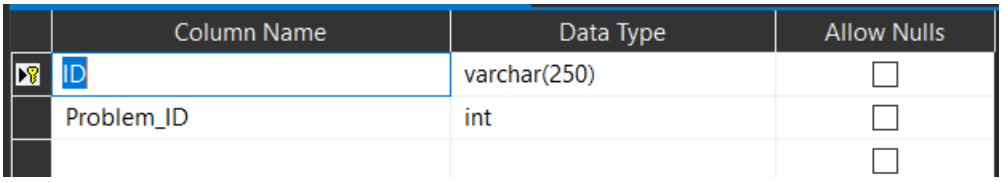
Language table:



Column Name	Data Type	Allow Nulls
ID	varchar(100)	☐
Input_ID	varchar(250)	☐
		☐

Figure 3.9 Language table

Input_Cases table:



Column Name	Data Type	Allow Nulls
ID	varchar(250)	☐
Problem_ID	int	☐
		☐

Figure 3.10 Input_Cases table

Contest table:

	Column Name	Data Type	Allow Nulls
	C_ID	int	☐
	Name	varchar(100)	☐
	Start_at	smalldatetime	☐
	End_in	smalldatetime	☐
	Admin_ID	int	☐
	visibility	bit	☐
			☐

Figure 3.11 Contest table

3.4.4 Use Case Diagram.

Figure 3.12 Authorized User

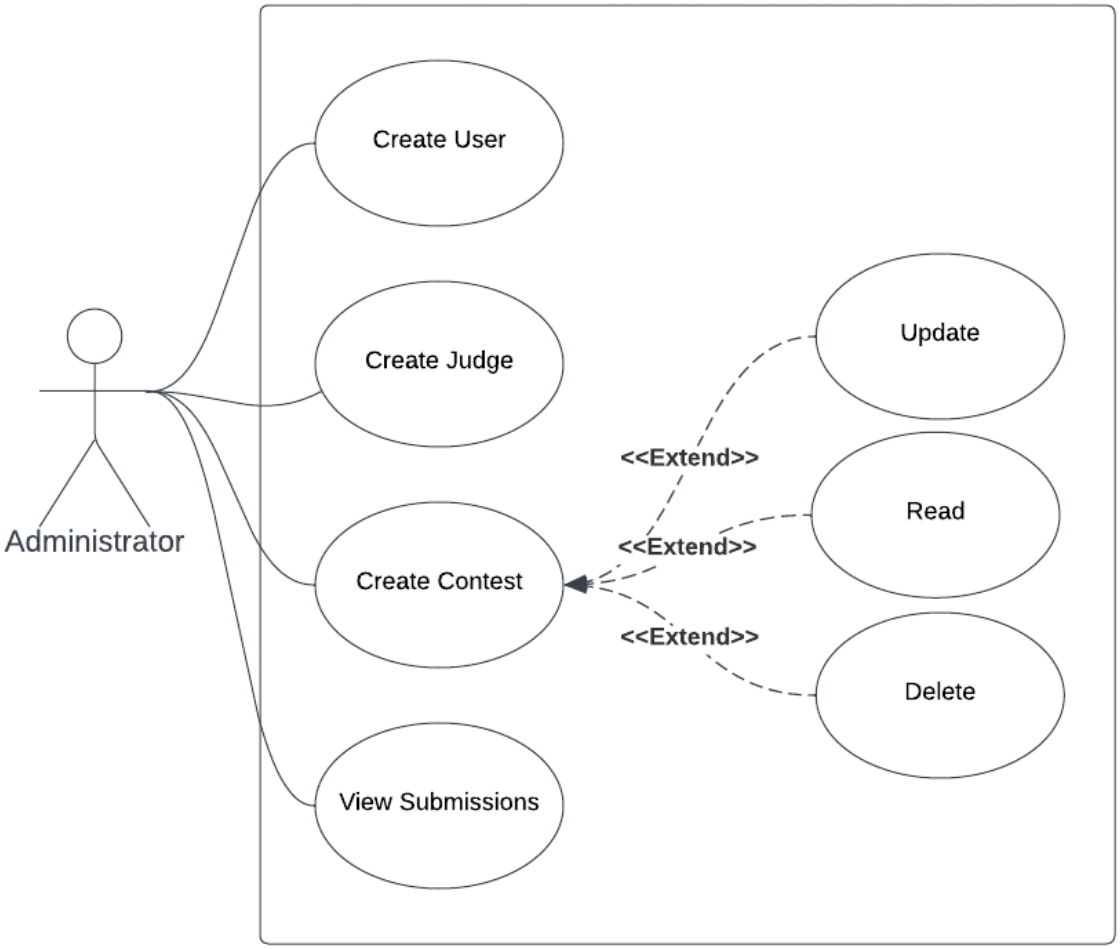


Figure 3.13 Admin User

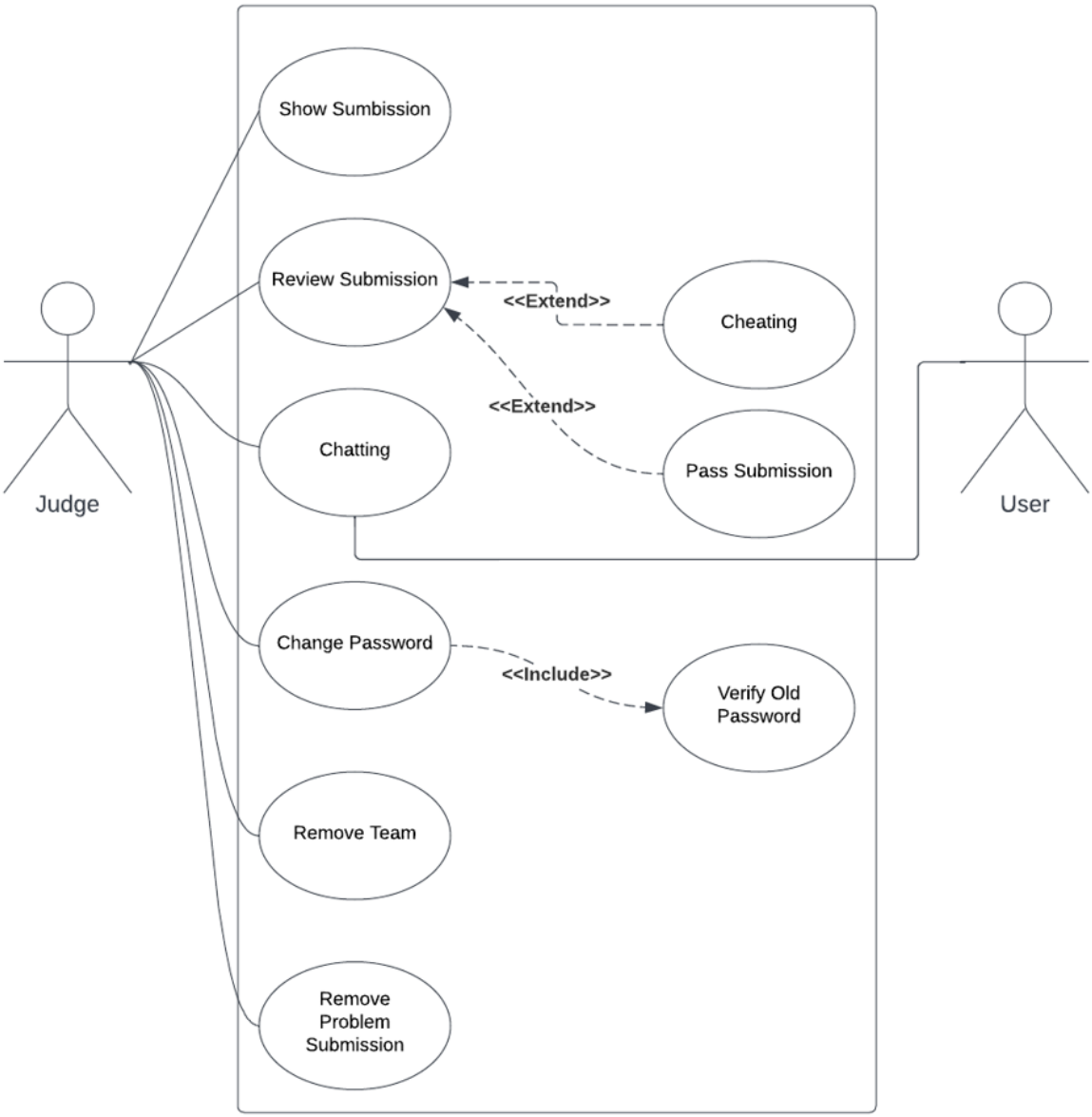


Figure 3.14 Judge

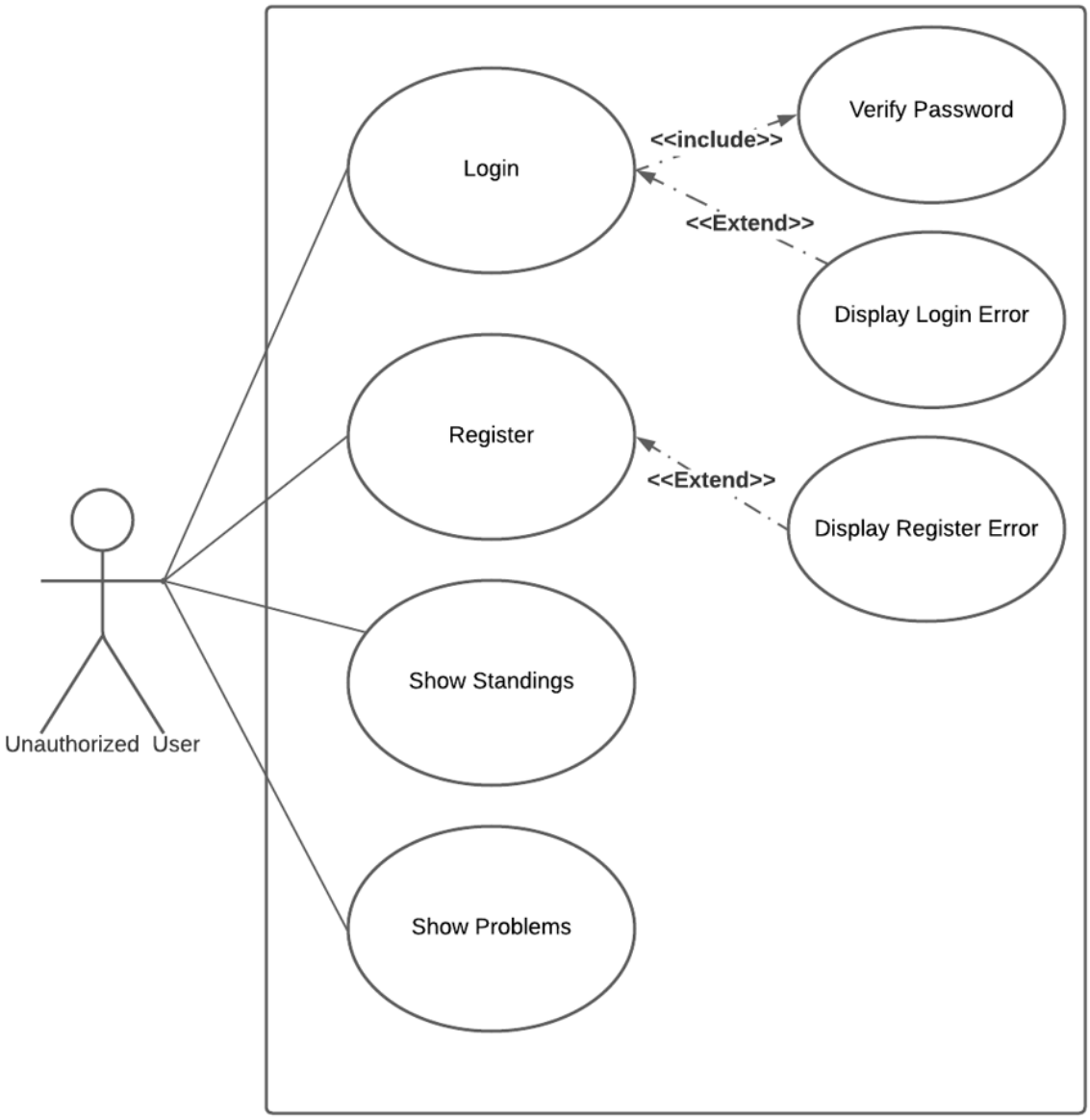


Figure 3.14 Unauthorized User

Chapter 4
Tools

4.1 Overview :

- Product Perspective :

  A Training application contains the following.

- Users :

  It include all the Users in the application .

- Admin :

  It include all the admins in the application.

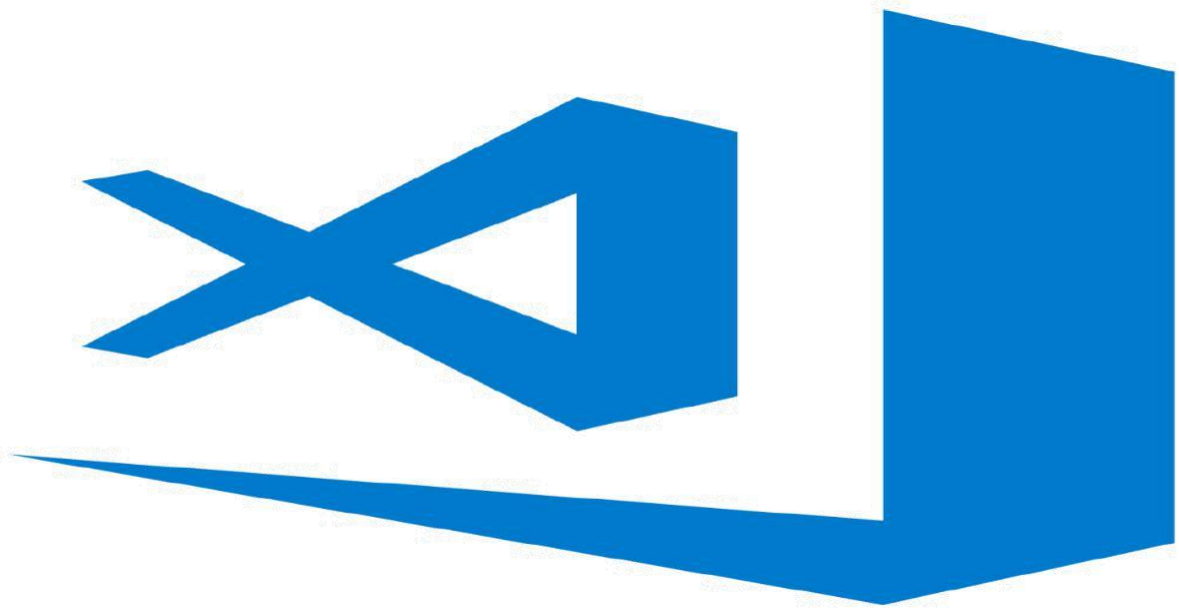
4.2 Application Function :

- By using this System/Application the CodeHex has to sign up/sign in so he can use all the features, he can choose the Contest that he wants then register to it, Also he can see all the Problems Available in the contest.

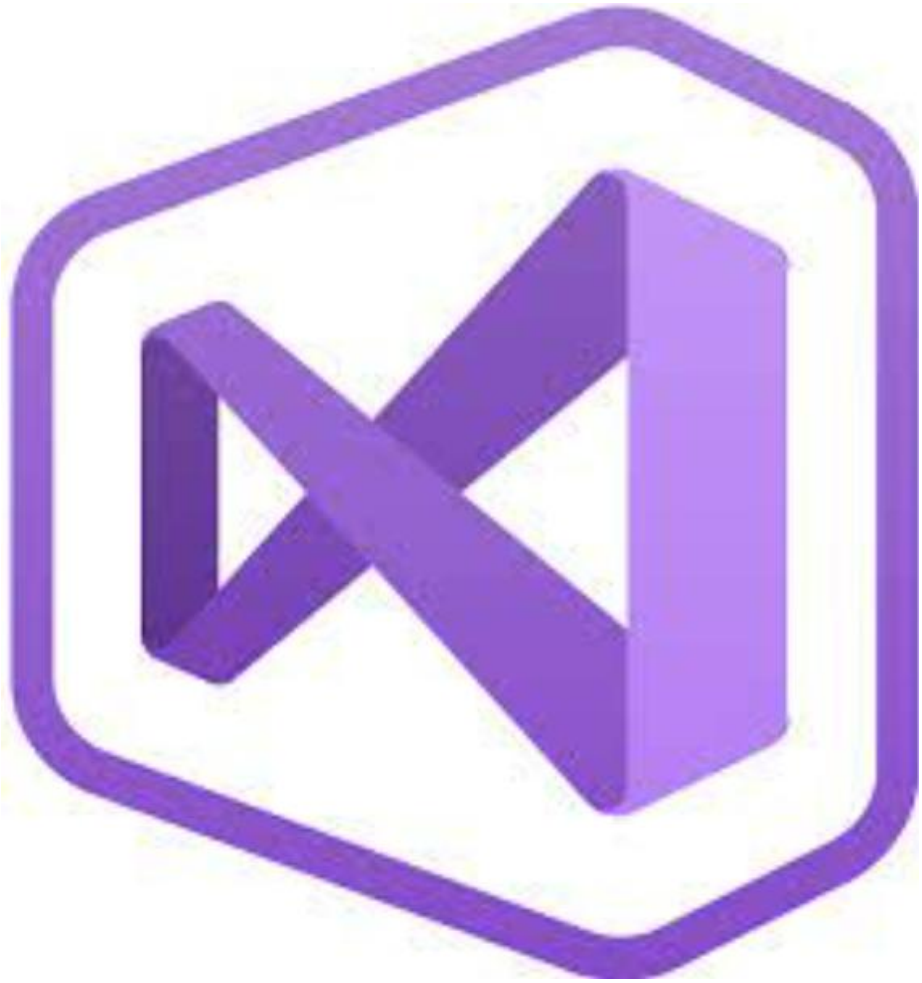
- By using this System/Application the Admin has to sign up/sign in so it can use all the features, he can, add his Contest that he wants then add Problems to it, Also he can see all the Submission and have full control on it in the application/system.

4.3 Main Work :

Visual Studio Code : Is a dual-licensed source-code editor made by Microsoft for Windows, Linux and macOS. In the Stack Overflow 2019 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool, with 50.7% of 87,317 respondents reporting that they use it. Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python and C++.It is based on the Electron framework, which is used to develop Node.js Web applications that run on the Blink layout engine. Visual Studio Code employs the same editor component (code named "Monaco") used in Azure DevOps (formerly called Visual Studio Online and Visual Studio Team Services).



4.4 Visual Studio : Visual Studio is a comprehensive integrated development environment (IDE) created by Microsoft for .NET developers. It provides a user-friendly interface and a wide range of tools that allow developers to create, debug, test, and deploy .NET applications. Some of the key features of Visual Studio include code editing and syntax highlighting, project management, version control integration, debugging tools, unit testing, profiling, and deployment options for desktop, web, and mobile platforms. Additionally, Visual Studio supports a variety of programming languages, including C#, F#, VB.NET, and more. Overall, Visual Studio is an essential tool for .NET developers looking to streamline their development workflows and increase productivity.



4.5 Website Content :

-Front-End :

• **HTML & CSS :** HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets) are two foundational technologies used to create and design web pages. HTML is a markup language used to structure content on the web page. It consists of a series of tags that are used to define the structure of the content, such as headings, paragraphs, images, links, and lists. HTML provides the basic structure and content of a web page. CSS, on the other hand, is a style sheet language used to define the visual appearance of the content on the web page. It consists of a series of rules that specify how the content should be displayed, such as the font, color, size, layout, and positioning of the various elements on the page. CSS allows web designers to create visually appealing and responsive designs that can adapt to different screen sizes and device types. Together, HTML and CSS are essential tools for creating modern, dynamic, and interactive websites. They are often used in conjunction with other technologies such as JavaScript to create rich web applications and user interfaces.

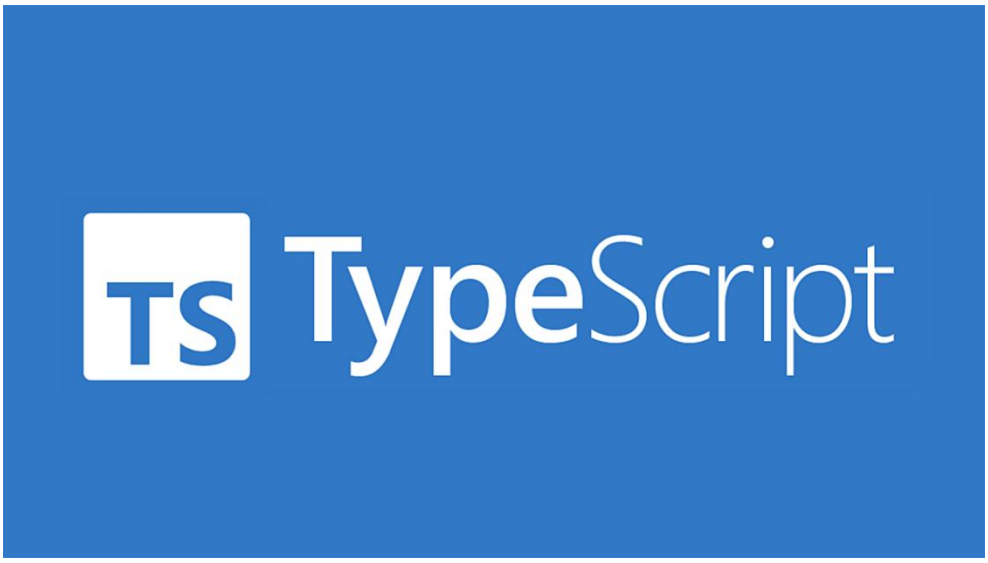
• **Typescript:** TypeScript is a programming language developed by Microsoft that is a superset of JavaScript. It is designed to address some of the shortcomings of JavaScript by adding optional static typing, class-based object-oriented programming, and other features that make it easier to write and maintain large-scale applications.

One of the key features of TypeScript is its support for static typing. This means that developers can specify the type of a variable, function, or object property at the time of declaration, and the TypeScript compiler will enforce those types throughout the codebase, catching errors at compile-time rather than at runtime. This can help reduce bugs and make code more predictable and easier to maintain.

TypeScript also supports class-based object-oriented programming, which allows developers to define classes with properties and methods, and create instances of those classes. This can help make code more modular, reusable, and easier to read.

In addition, TypeScript includes a number of other features that are not present in JavaScript, such as interfaces, enums, generics, decorators, and more. These features can help make code more expressive and easier to reason about.

Overall, TypeScript is a powerful and flexible language that can help developers write better, more maintainable code, particularly for larger-scale applications. It is often used in conjunction with popular web development frameworks such as Angular, React, and Vue.js.



• **Angular :** Angular is an open-source web application framework developed and maintained by Google. It is a complete solution for building dynamic, single-page web applications (SPAs) using HTML, CSS, and TypeScript.

Angular provides a set of powerful tools and features that help developers build complex, data-driven applications with ease. These include:

1- Component-based architecture: Angular applications are built using components, which are self-contained, reusable UI elements that can be combined to create complex interfaces.

2- Two-way data binding: Angular provides a powerful data binding system that allows changes in the UI to be automatically reflected in the application data, and vice versa.

3- Dependency injection: Angular's dependency injection system makes it easy to manage dependencies between components and services, and helps keep code modular and testable.

4- Routing: Angular's built-in routing system allows developers to create multiple views and navigate between them without reloading the entire page.

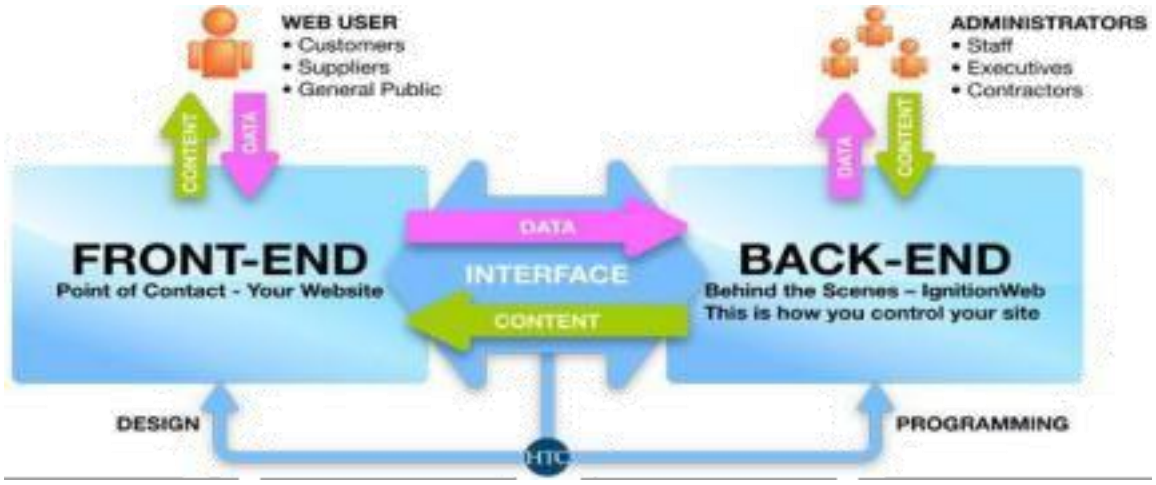
5- Forms: Angular provides a powerful forms module that makes it easy to build complex forms with validation, custom inputs, and more.

6- Reactive programming: Angular provides support for reactive programming using RxJS, making it easy to work with asynchronous data streams.

Angular is widely used for building enterprise-scale applications, and is supported by a large and active community of developers. It integrates well with other popular web technologies such as TypeScript, HTML, CSS, and various back-end frameworks, making it a versatile and powerful tool for building modern web application.

- **Back-End :**

The backend of a website is the place that contains all the data and relevant information that is to be shown to the visitors with the help of a browser. The frontend of a website is merely how the information is presented to the users, and it fetches everything from the backend to display in user browsers. The image below shows this concept visually,



Backend developer focuses on databases, scripting, and the architecture of websites. Code written by back-end developers helps to communicate the database information to the browser. A backend developer works with the following:-

- Web Development Languages
- Database and cache
- Server
- API (REST & SOAP)

Development Languages Backend engineer should know at least one server-side programming languages like PHP, Java, Python, Ruby, . Net etc.

● What is C#?

- C# (pronounced "see sharp") is a modern, high-level programming language developed by Microsoft in the early 2000s. It was designed to be a simple, yet powerful language that could be used to write a wide range of applications for Windows and other platforms.

- C# is an object-oriented language, which means that it allows developers to create classes and objects that encapsulate data and behavior. It also features a strong type system, which helps catch errors at compile time rather than runtime, and automatic memory management through garbage collection.

- C# is widely used for developing desktop applications, web applications, games, mobile apps, cloud services, and more. It is also frequently used in conjunction with the .NET Framework, a software development platform created by Microsoft that provides tools and libraries for building a wide range of applications.

- Overall, C# is a versatile and powerful programming language that has gained popularity among developers due to its ease of use, robustness, and extensive support from Microsoft and the wider developer community.

❖ Coding in C# look like :

- Coding in C# involves writing instructions in the C# programming language with the goal of creating software applications. Like many programming languages, C# is used to create code that can be executed by a computer, and the language has specific syntax rules and structure that must be followed in order for the code to work properly.

- In C#, developers create classes that contain data and behavior relevant to a particular aspect of the application being built. These classes can inherit from other classes or interfaces, and they can be instantiated as objects to perform specific tasks within the application.

- C# also includes various control structures such as conditional statements, loops, and exception handling mechanisms that allow developers to control how their code executes under different conditions. In addition, C# supports features such as lambda expressions, LINQ queries, and asynchronous programming that allow developers to write more concise and efficient code.

- Overall, coding in C# requires a strong understanding of programming principles and techniques, as well as familiarity with the specific syntax and structure of the C# language. It also often involves testing, debugging, and refining code to ensure it produces the desired outcome and functions correctly within the larger context of the application being developed.

- **MSQL Server and SSMS:**

MSql Server : Microsoft SQL Server is a relational database management system (RDBMS) developed by Microsoft. It is used to store and retrieve data as requested by other software applications, which may run on the same computer or on another computer across a network.

- SQL Server is designed to be scalable, meaning that it can handle large amounts of data and users without sacrificing performance. It also includes features for security, backup and recovery, and business intelligence.

- SQL Server supports several programming languages, including SQL, Transact-SQL (T-SQL), and .NET languages such as C# and Visual Basic. It can be used with a variety of development tools, such as Visual Studio and SQL Server Management Studio.

Some of the key features of SQL Server include:

- Built-in support for high availability and disaster recovery
- Integration with Microsoft's cloud services, such as Azure
- Advanced analytics and machine learning capabilities
- In-memory processing for improved performance
- Integration with other Microsoft products, such as Excel and SharePoint

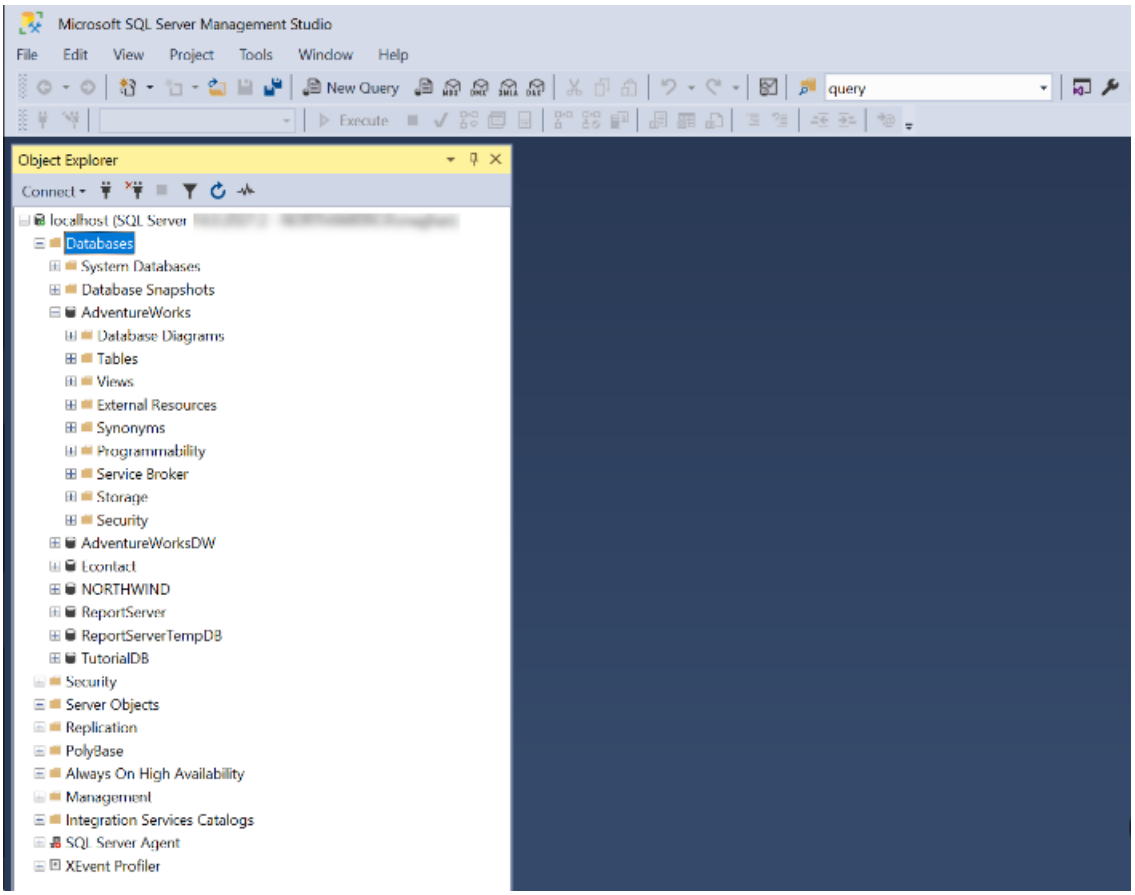
• **Xml :** Basically , Xml is used for layout designing. All the UI and layout of your app is designed using xml. Unlike Java (which is Back Bone of your app), xml helps you to design your app , how it will look , how components like buttons , textview , etc will be placed and their styling. Apart from these , xml is also used for parsing data either from database or server into your android app.(Xml parsing).

- **SSMS:** SSMS stands for SQL Server Management Studio, which is a software application that is used to manage Microsoft SQL Server databases. It provides a graphical user interface (GUI) for managing and developing SQL Server databases, as well as tools for writing and executing SQL queries.

With SSMS, you can perform a wide range of tasks, including creating, modifying, and deleting database objects such as tables, views, and stored procedures, as well as managing security, backups, and other administrative tasks. You can also use it to import and export data, execute scripts, and analyze query performance.

One of the key benefits of using SSMS is that it allows you to work with SQL Server in a more efficient and streamlined way, particularly when working on complex projects or large databases. It can help you save time and reduce errors by providing a visual interface for many common tasks, and by automating some of the more tedious aspects of SQL Server management.

In summary, SSMS is a powerful tool for managing and developing SQL Server databases, and it's widely used by developers, database administrators, and other IT professionals who work with SQL Server on a regular basis.



- **.NET Core FrameWork :**

.NET Core is a free and open-source, cross-platform framework for building modern applications. It was developed by Microsoft and released in 2016. The framework is designed to be modular, lightweight, and highly scalable, making it ideal for building cloud-based applications and microservices.

.NET Core includes a rich set of libraries and tools that make it easy to develop and deploy applications on a variety of platforms, including Windows, Linux, and macOS. The framework supports multiple programming languages, including C#, F#, and Visual Basic .NET, and provides a variety of application development models, such as Web API, MVC, and Razor Pages.

Some of the key features of .NET Core include high performance, native support for containers, automatic memory management, and a flexible deployment model. Additionally, .NET Core is fully supported by Microsoft, with regular updates and security patches provided through its extensive support and developer community.

Chapter 5
System Implementation

5.1 Web Design :-

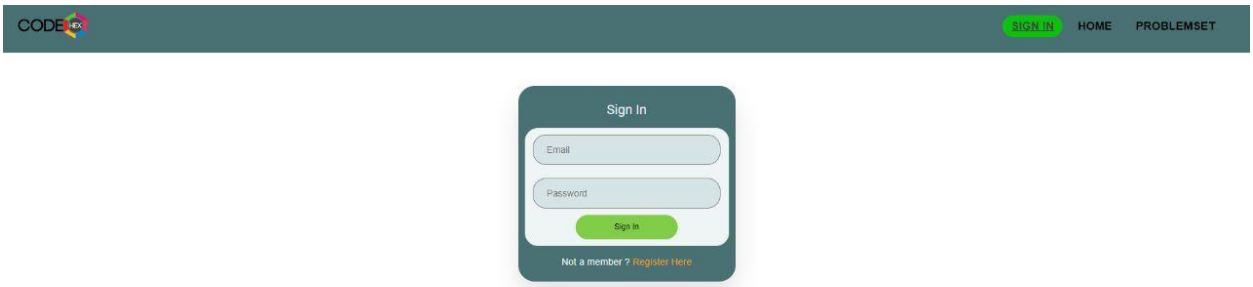


Figure 5.1 Login page

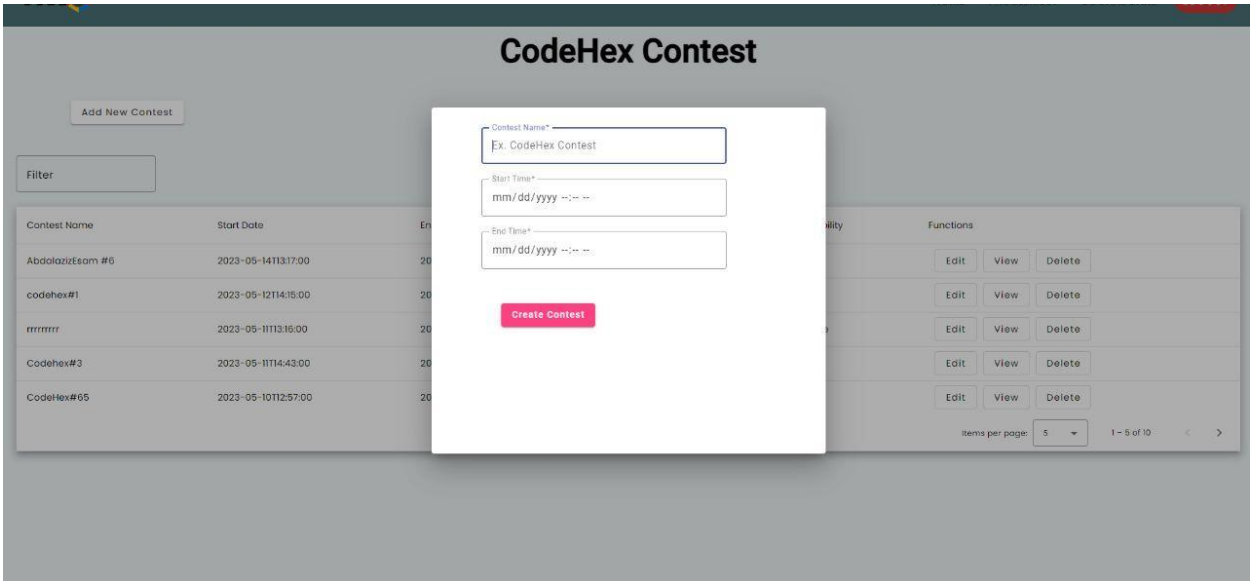


Figure 5.2 Create Contest

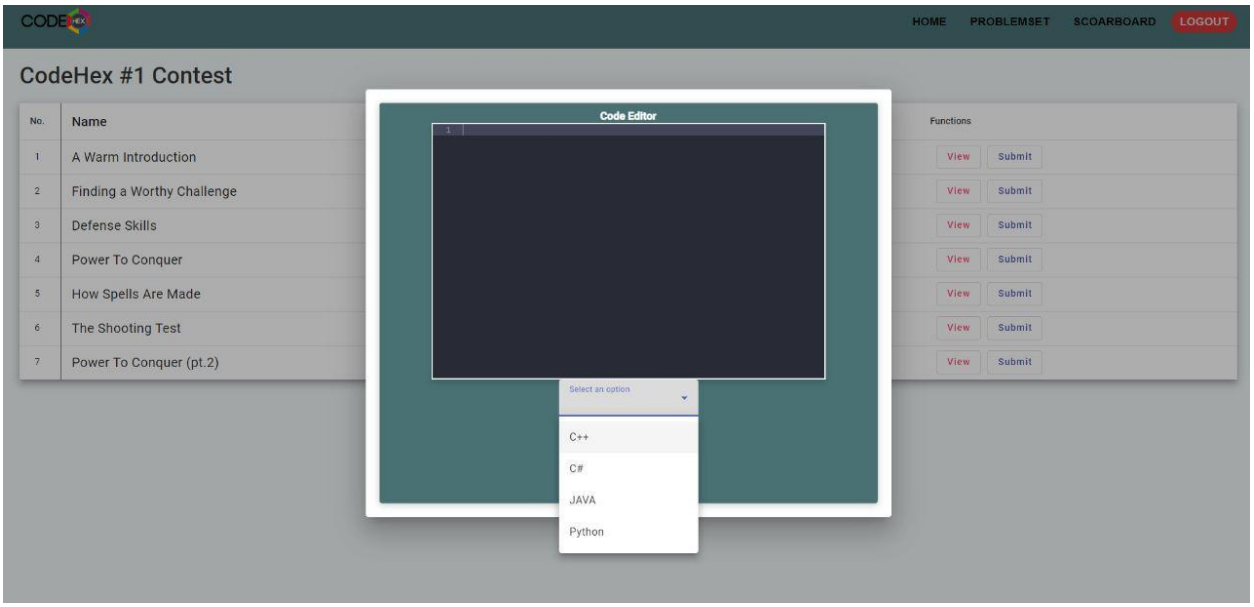


Figure 5.3 Submission

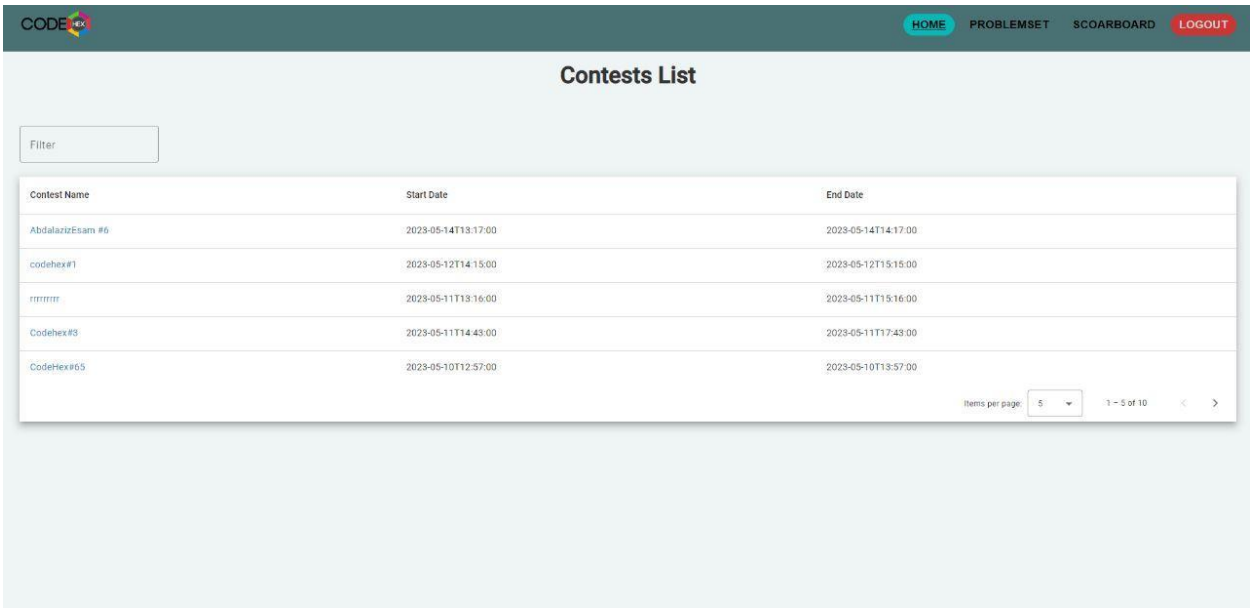


Figure 5.4 List of Contests

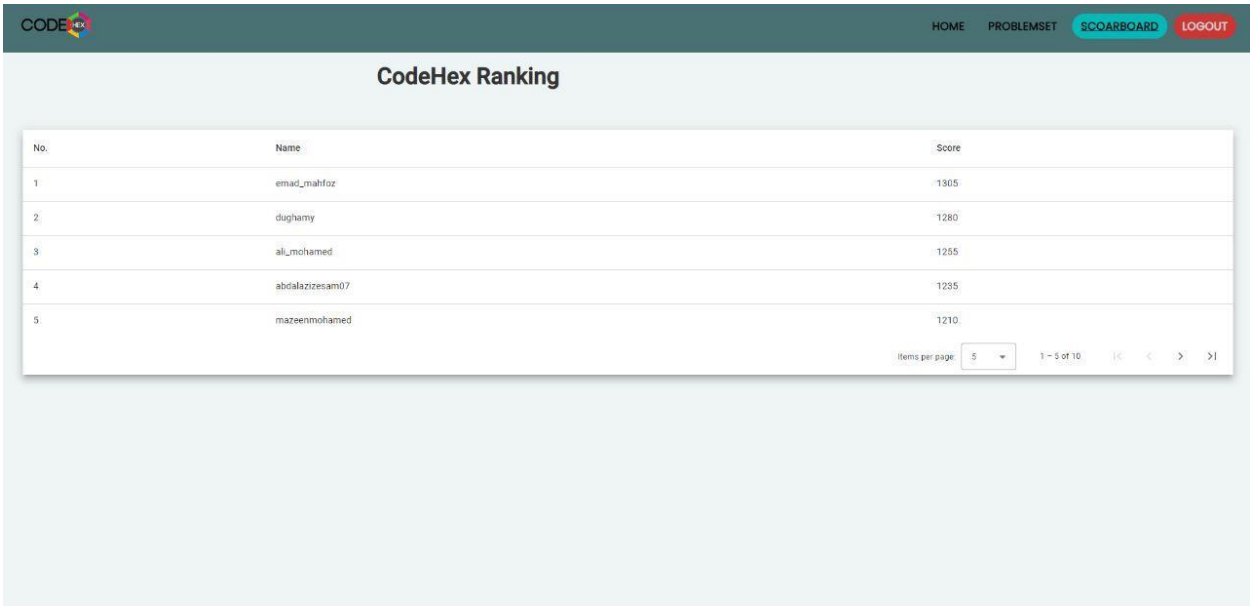


Figure 5.5 Scoreboard

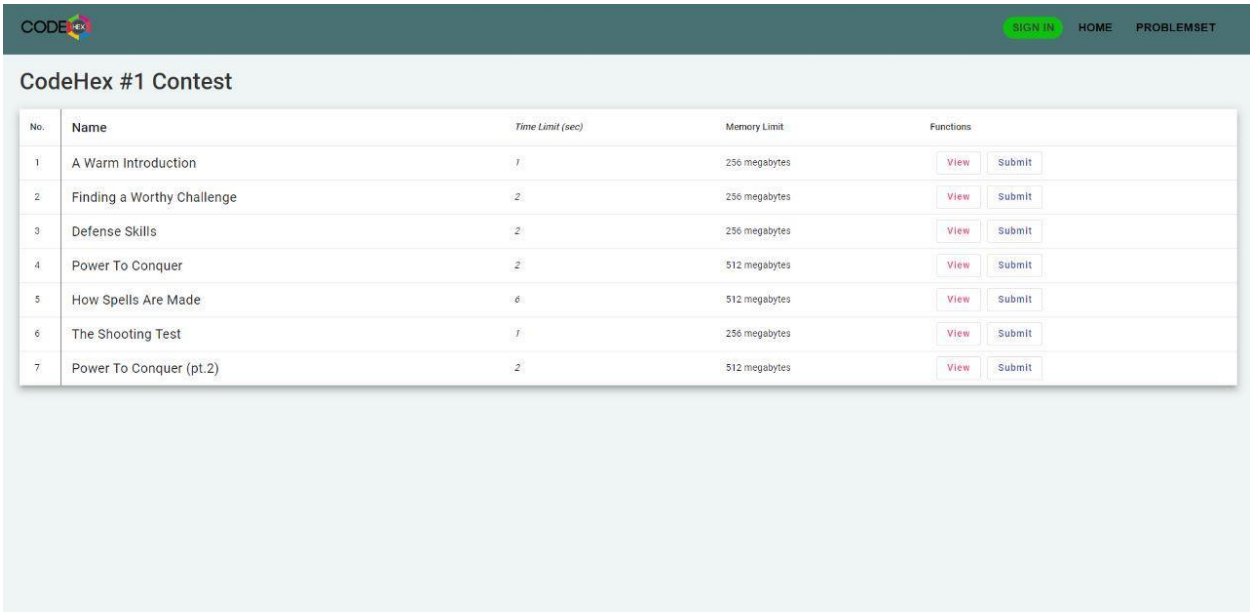


Figure 5.6 ProblemSet

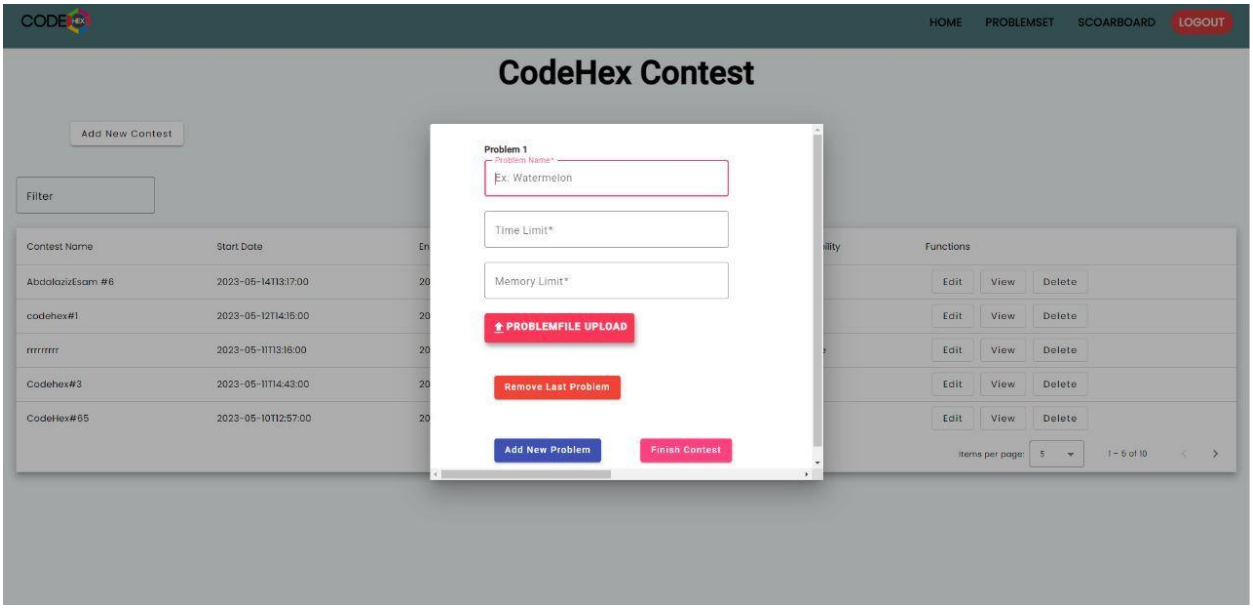


Figure 5.7 Contest Form

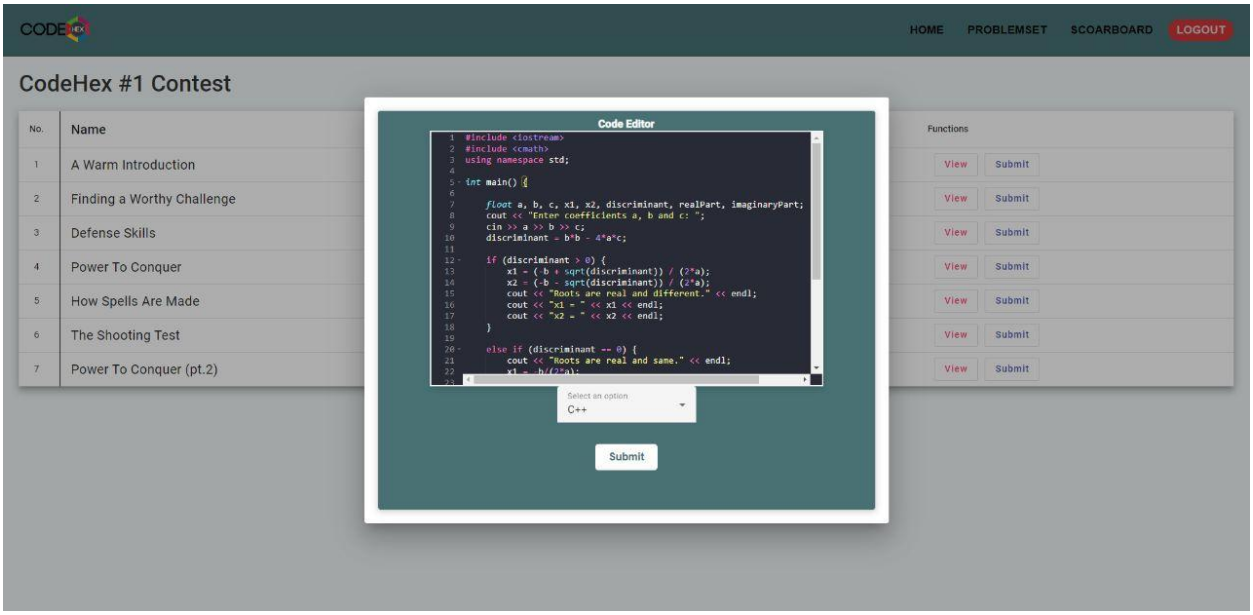


Figure 5.8 Code Highlight

5.3 Code Snippets :-

```
    /// <summary>
    /// For login user in generale
    /// He can login by Name or Email like Codeforces
    /// </summary>
    /// <returns>If data correct 204 ; If data is wrong will return 401 and Json file with the wrong fileds. </returns
    [HttpPost(template: "login")]
    0 references
    public async Task<IActionResult> login([FromBody] LoginBinding item)
    {
        if (item == null)
        {
            return StatusCode(statusCode: (int)HttpStatusCode.BadRequest);
        }

        bool Valid = await AccountValidate.IsVaildAccount(UserName: item.Email, password: item.Password, HttpContext);

        if (Valid)
        {
            return StatusCode(statusCode: (int)HttpStatusCode.OK);
        }
        else
        {
            ModelState.AddModelError(key: "Email", errorMessage: "Email or Password is wrong");
            var IEnumerable<ModelError> Errors = ModelState.Values.SelectMany(selector: ModelStateEntry v => v.Errors);

            return StatusCode(statusCode: (int)HttpStatusCode.Unauthorized, value: Errors);
        }
    }
    ...
/// <summary>
/// Make Cookie for user
/// </summary>
/// <returns>return one if Account Data is Valid or Zero if Data isn't Valid</returns>
1 reference
static public async Task<bool> IsVaildAccount(string UserName, string password, HttpContext HttpContext)
{
    using (var ProjectDbContext db = new ProjectDbContext())
    {
        var User User = db.Users
        .FirstOrDefault(predicate: User M => (M.Email == UserName || M.Name == UserName)
        && M.Password == password);

        if (User != null && User.IsValid == 1 && User.CheatTimes < 3)
        {
            var List<Claim> claims = new List<Claim>{
                        new Claim(type:"ID", value: User.Id.ToString()),
                        new Claim(type: ClaimTypes.Name, value: User.UserName),
                        new Claim(type: ClaimTypes.Role, value: db.Roles.Find(keyValues: User.RId).Name)
            };
            var ClaimsIdentity identity = new ClaimsIdentity(claims, authenticationType: CookieAuthenticationDefaults.AuthenticationScheme);
            await HttpContext.SignInAsync(
            scheme: CookieAuthenticationDefaults.AuthenticationScheme,
            principal: new ClaimsPrincipal(identity),
            properties: new AuthenticationProperties
            {
                IsPersistent = true,
                AllowRefresh = true
            });

            return true;
        }
        return false;
    }
}
```

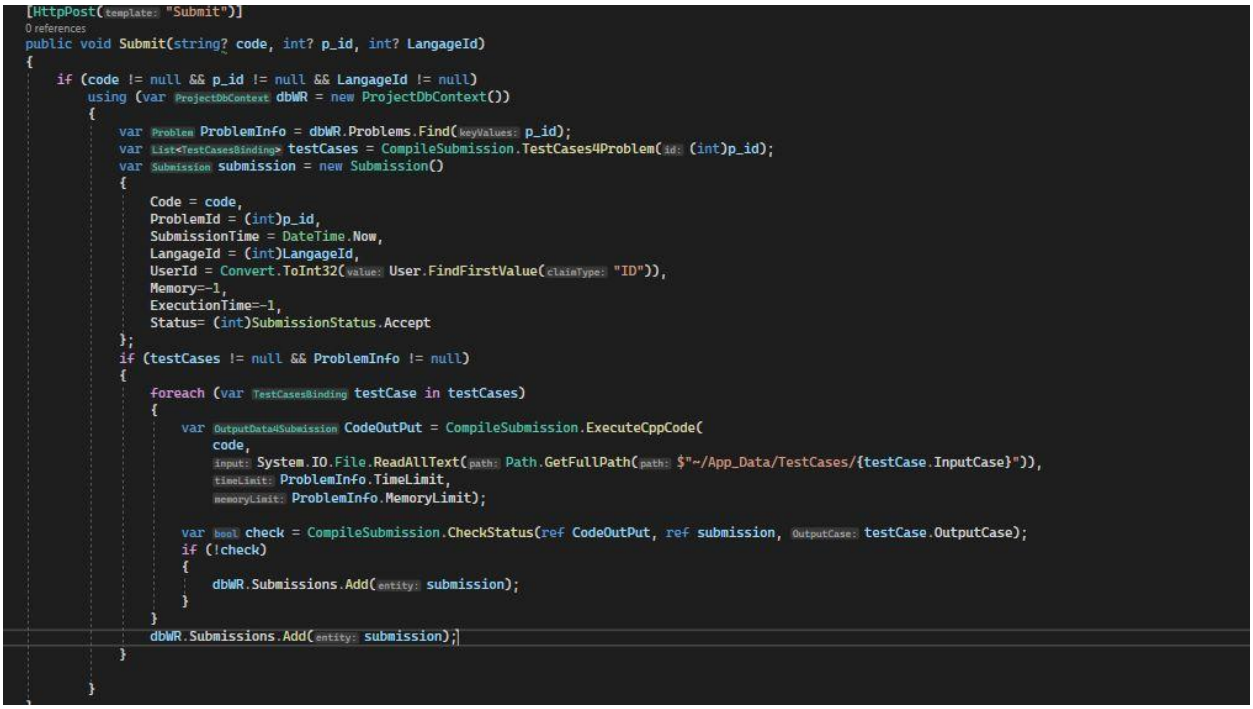
Figure 5.9 Login Codes

Login code using POST method to get the data entered in the login form then checks if it matches the data in the database to verify the login information.

```
1 reference
static public OutputData4Submission ExecuteCppCode(string code, string input, decimal timeLimit, int memoryLimit)
{
    var OutputData4Submission outputData = new OutputData4Submission();
    string sourceFile = "source.cpp", executableFile = "executable.exe";
    File.WriteAllText(path: sourceFile, contents: code);
    var ProcessStartInfo startInfo = new ProcessStartInfo
    {
        FileName = "g++",
        Arguments = $"{Path.Combine(".", sourceFile)} -o {Path.Combine(".", executableFile)}",
        UseShellExecute = false,
        RedirectStandardOutput = true,
        RedirectStandardError = true
    };
    var Process compilerProcess = Process.Start(startInfo);
    compilerProcess.WaitForExit();
    if (compilerProcess.ExitCode != 0)
    {
        outputData.Result = "Compilation error: " + compilerProcess.StandardError.ReadToEnd();
        return outputData;
    }
    startInfo.FileName = Path.Combine(".", executableFile);
    startInfo.Arguments = "";
    startInfo.UseShellExecute = false;
    startInfo.RedirectStandardOutput = true;
    startInfo.RedirectStandardError = true;
    startInfo.RedirectStandardInput = true;
    var Process executorProcess = new Process();
    executorProcess.StartInfo = startInfo;
    executorProcess.EnableRaisingEvents = true;
    executorProcess.StartInfo.EnvironmentVariables[key: "RLIMIT_AS"] = $"{memoryLimit}M";
    executorProcess.StartInfo.EnvironmentVariables[key: "RLIMIT_CPU"] = $"{timeLimit}";
    executorProcess.Start();
    executorProcess.StandardInput.Write(value: input);
    while (!executorProcess.HasExited)
    {
        while (!executorProcess.WaitForExit(milliseconds: 100))
        {
            long memoryUsage = executorProcess.WorkingSet64;
            if (memoryUsage >= memoryLimit * 1024L * 1024L)
            {
                executorProcess.Kill();
                outputData.Result = "Memory limit exceeded";
                outputData.memory = executorProcess.WorkingSet64 / 1024L;
                outputData.time = executorProcess.TotalProcessorTime.Milliseconds;
                return outputData;
            }
        }
        timeLimit -= 0.1M;
        if (timeLimit <= 0)
        {
            executorProcess.Kill();
            outputData.Result = "Time limit exceeded";
            outputData.memory = executorProcess.WorkingSet64 / 1024L;
            outputData.time = executorProcess.TotalProcessorTime.Milliseconds;
            return outputData;
        }
    }
    outputData.Result = Regex.Replace(input: executorProcess.StandardOutput.ReadToEnd(), pattern: @"\s+", replacement: " ");
    outputData.memory = executorProcess.WorkingSet64 / 1024L;
    outputData.time = executorProcess.TotalProcessorTime.Milliseconds;
    return outputData;
}
```

Figure 5.10 Run Code

Run solution Code for users in problem by verifying constraints of the problem from timelimit, memorylimit, correct syntax and…. etc .



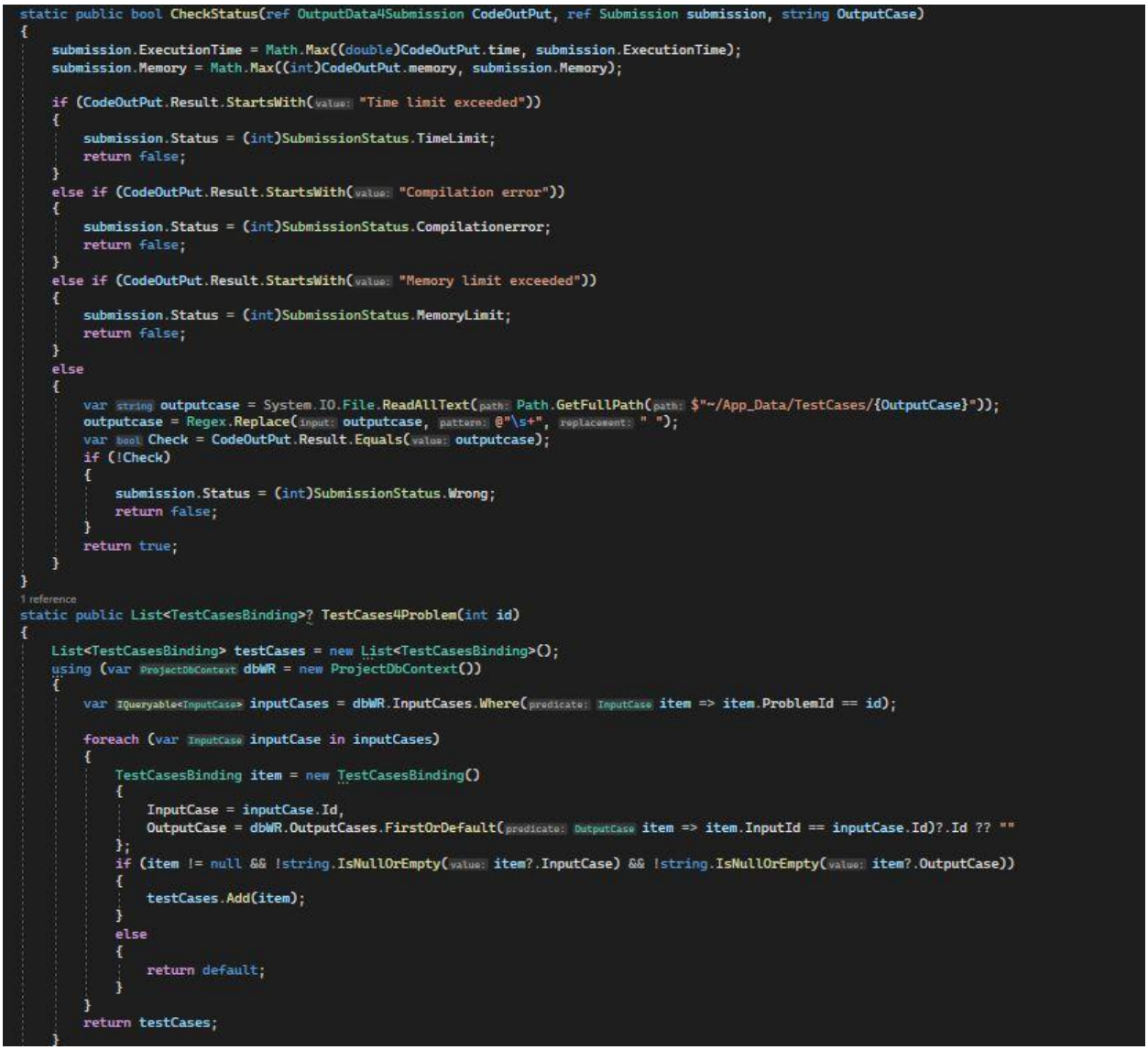
```
[HttpPost(template: "Submit")]
0 references
public void Submit(string? code, int? p_id, int? LangageId)
{
    if (code != null && p_id != null && LangageId != null)
        using (var ProjectDbContext dbWR = new ProjectDbContext())
        {
            var Problem ProblemInfo = dbWR.Problems.Find(keyValues: p_id);
            var List<TestCasesBinding> testCases = CompileSubmission.TestCases4Problem(id: (int)p_id);
            var Submission submission = new Submission()
            {
                Code = code,
                ProblemId = (int)p_id,
                SubmissionTime = DateTime.Now,
                LangageId = (int)LangageId,
                UserId = Convert.ToInt32(value: User.FindFirstValue(claimType: "ID")),
                Memory=-1,
                ExecutionTime=-1,
                Status= (int)SubmissionStatus.Accept
            };
            if (testCases != null && ProblemInfo != null)
            {
                foreach (var TestCasesBinding testCase in testCases)
                {
                    var OutputData4Submission CodeOutPut = CompileSubmission.ExecuteCppCode(
                        code,
                        input: System.IO.File.ReadAllText(path: Path.GetFullPath(path: $"~/App_Data/TestCases/{testCase.InputCase}")),
                        timeLimit: ProblemInfo.TimeLimit,
                        memoryLimit: ProblemInfo.MemoryLimit);

                    var bool check = CompileSubmission.CheckStatus(ref CodeOutPut, ref submission, OutputCase: testCase.OutputCase);
                    if (!check)
                    {
                        dbWR.Submissions.Add(entity: submission);
                    }
                }
                dbWR.Submissions.Add(entity: submission);
            }

        }
```

Figure 5.11 Submit solution

Get a solution for specify problem .



```
static public bool CheckStatus(ref OutputData4Submission CodeOutPut, ref Submission submission, string OutputCase)
{
    submission.ExecutionTime = Math.Max((double)CodeOutPut.time, submission.ExecutionTime);
    submission.Memory = Math.Max((int)CodeOutPut.memory, submission.Memory);

    if (CodeOutPut.Result.StartsWith(value: "Time limit exceeded"))
    {
        submission.Status = (int)SubmissionStatus.TimeLimit;
        return false;
    }
    else if (CodeOutPut.Result.StartsWith(value: "Compilation error"))
    {
        submission.Status = (int)SubmissionStatus.Compilationerror;
        return false;
    }
    else if (CodeOutPut.Result.StartsWith(value: "Memory limit exceeded"))
    {
        submission.Status = (int)SubmissionStatus.MemoryLimit;
        return false;
    }
    else
    {
        var string outputcase = System.IO.File.ReadAllText(path: Path.GetFullPath(path: $"~/App_Data/TestCases/{OutputCase}"));
        outputcase = Regex.Replace(input: outputcase, pattern: @"\s+", replacement: " ");
        var bool Check = CodeOutPut.Result.Equals(value: outputcase);
        if (!Check)
        {
            submission.Status = (int)SubmissionStatus.Wrong;
            return false;
        }
        return true;
    }
}
1 reference
static public List<TestCasesBinding>? TestCases4Problem(int id)
{
    List<TestCasesBinding> testCases = new List<TestCasesBinding>();
    using (var ProjectDbContext dbWR = new ProjectDbContext())
    {
        var IQueryable<InputCase> inputCases = dbWR.InputCases.Where(predicate: InputCase item => item.ProblemId == id);

        foreach (var InputCase inputCase in inputCases)
        {
            TestCasesBinding item = new TestCasesBinding()
            {
                InputCase = inputCase.Id,
                OutputCase = dbWR.OutputCases.FirstOrDefault(predicate: OutputCase item => item.InputId == inputCase.Id)?.Id ?? ""
            };
            if (item != null && !string.IsNullOrEmpty(value: item?.InputCase) && !string.IsNullOrEmpty(value: item?.OutputCase))
            {
                testCases.Add(item);
            }
            else
            {
                return default;
            }
        }
        return testCases;
    }
```

Figure 5.12 Get Test Case & Code Validation

Get test case input and output and use it for test code by comparing output of input case to code when run **Figure 5.16**.

```
[HttpPut(template: "contest/{id:int}")]
0 references
public async Task<ActionResult> Contest([FromRoute] int? id, [FromBody] ContestBindingModel Item)
{
    await ContestServices.ValidName(Name: Item.Name, ModelState);
    await ContestServices.ValidDates(Start_at: Item.Start_at, End_in: Item.End_in, ModelState);
    if (id == null || Item.ID == null || id != Item.ID)
    {
        ModelState.AddModelError(key: "", errorMessage: "UnValid data");
    }
    if (!ModelState.IsValid)
    {
        var IEnumerable<ModelError> Errors = ModelState.Values.SelectMany(selector: ModelStateEntry v => v.Errors);

        return StatusCode(statusCode: (int)HttpStatusCode.Conflict, value: new { Errors, Item });
    }
    int IsUpdated = await ContestServices.UpdateContest(Item);

    if (IsUpdated != 1)
    {
        return StatusCode(statusCode: (int)HttpStatusCode.InternalServerError);
    }
    else
    {
        return StatusCode(statusCode: (int)HttpStatusCode.NoContent);
    }
}
```

Figure 5.13 Edit contest

```
/// <summary>
/// Change State Contest
/// </summary>
/// <returns>return 204 if operation work corretly or 500 if server error</returns>
[HttpDelete(template: "contest/{id:int}")]
0 references
public async Task<ActionResult> Contests([FromRoute] int? id)
{
    ProjectDbContext dbWR = new ProjectDbContext();
    if (!(id is null))
    {
        var Contest contest = dbWR.Contests.FirstOrDefault(predicate: Contest cntest => cntest.CId == id);

        if (!(contest is null))
        {
            contest.Visibility = !(contest.Visibility ?? true && true);
            try
            {
                await dbWR.SaveChangesAsync();
            }
            catch
            {
                return StatusCode(statusCode: (int)HttpStatusCode.InternalServerError);
            }
        }
    }
    return StatusCode(statusCode: (int)HttpStatusCode.NoContent);
}
```

Figure 5.14 Delete contest

Delete Contest by admin by hiding it from users .

```
!pip install -U sentence-transformers

from sentence_transformers import SentenceTransformer

model = SentenceTransformer('sentence-transformers/all-MiniLM-L6-v2')
sentences = ["apple", "app", "orange", "or"]
embeddings = model.encode(sentences)

import numpy as np

def cos_sim(M):
  norms=np.linalg.norm(M,axis=1).reshape(len(M),1)
  return np.matmul(M,M.T)/np.matmul(norms,norms.T)

cos_sim(embeddings)
```

Figure 5.15 Similarity checker

Use python code for validate the submission and there is not cheaters .

Chapter 6
Conclusions & Future Work

6.1 Conclusion:

The benefits of using CodeHex include automating many of the tasks involved in organizing a contest, reducing the workload on organizers and saving time, providing real-time feedback to participants, enabling them to adjust their strategies accordingly, and improving the fairness and transparency of the contest, In addition to the benefits that CodeHex provides to organizers and event managers, it also has several advantages for students who participate in the contests. With features such as real-time feedback, centralized submission and evaluation of entries, and transparent scoring system, students can receive valuable feedback on their performance, track their progress, and learn from their mistakes. Additionally, students can communicate with other participants and organizers through the platform, fostering collaboration, networking, and skill-building opportunities.

Overall, CodeHex has the potential to enhance the experience of both participants and organizers, making it easier to run successful contests while minimizing the risk of errors or disputes. As technology continues to evolve, this system has endless possibilities for innovation and expansion in enhancing the management of contests. The CodeHex contest management system not only streamlines the administration of contests but also promotes student engagement and learning, making it an ideal platform for organizations looking to offer engaging and interactive events for their participants.

6.2 Future Work:

• Integration with external platforms: You could explore the possibility of integrating CodeHex with popular platforms like GitHub or GitLab to enable participants to submit their code directly from these platforms.

• Improved scoring algorithms: Currently, CodeHex supports score tracking, but you could consider developing more sophisticated algorithms that take into account the difficulty of the problems and the time taken by participants to solve them.

• Gamification features: To enhance engagement among participants, you could develop gamification features such as badges, leaderboards, or achievements.

• User feedback and analytics: You could implement user feedback mechanisms to gather feedback on the system's usability, design, and functionality. Additionally, you could use analytics tools to track usage patterns and identify areas for improvement.

• Multi-language support: Currently, CodeHex only supports a single programming language. You could consider adding support for multiple languages to enable a wider range of participants to join the contests.

• Customizable themes and templates: To improve the branding and customization options, you could add customizable themes and templates to CodeHex, allowing organizers to tailor the look and feel of the system to match their brand identity.

• Mobile application development: With an increasing number of users accessing web applications on mobile devices, you could consider developing a mobile application version of CodeHex to make it more accessible and user-friendly on mobile devices.

References :-

[1] W3Schools Online Web Tutorials. Last Visited [20/12/2022]

[2] www.stackoverflow.com. Last Visited [2/1/2022]

[3] www.developer.mozilla.org. Last Visited [15/1/2023]

[4] Software Life Cycle available from:

(http://en.wikipedia.org/wiki/Systems_Development_Life_Cycle).
Last Visited [20/12/2020]

[5] http://en.wikipedia.org/wiki/Software_testing. Last Visited [20/12/2023]

[6] Jeremy Keith, “HTML5 for Web Designers”, 2011.

[7] Dan Cederholm, “CSS3 for Web Designers”, 2010.

[8] http://en.wikipedia.org/wiki/Documentation. Last Visited [27/4/2020]

[9] Douglas Crockford, “JavaScript: The Good Parts”, 2008.

[10] Mark Boulton, “A Practical Guide to Designing for the Web”, 2009.

[11] Yiyi Sun, “Practical Application Development with AppRun”, 2019.

[12] Async Await and the Generated StateMachine - CodeProject .

Last Visited [22/5/2023]

[13] Indexing in Sql (E-book) . Last Visited [28/5/2023]

[14] C# Corner - Community of Software and Data Developers .

Last Visited [10/5/2023]