Cairo University
Faculty of Computers and Artificial intelligence
Department of Computer Sciences

Groofy Code

Supervised by

Dr. Mohamad Abdelwahab

TA. ##########

Implemented by

20200136	*Hazem Adel Khalel Nabawy*
20200359	*Omar Mohmed Mostafa*
20200268	*Doha Abdelbasset Ahmed*
20200514	*Marwa Ahmed Mohamed Mubarak*
20200501	*Mahmoud Abdelrady Gad*

Graduation Project
Academic Year 2023-2024
Midyear Short Documentation

Chapter 1: Introduction

Motivation (Abstract)

In the current landscape of online coding platforms, enthusiasts often face a fragmented experience, with platforms focusing primarily on coding challenges but lacking a cohesive ecosystem for diverse coding interests. Groofy Code emerges as a solution to this problem by integrating challenges, problem-solving, and competitive 1 vs 1 matches within a comprehensive framework. Existing platforms often lack an inclusive social aspect, hindering collaboration and community building among coding enthusiasts.

Our motivation to develop Groofy Code stems from a passion for fostering a collaborative and engaging environment for coding enthusiasts. We aim to provide a unified platform where users can seamlessly transition between challenges, problem-solving activities, and competitive matches while building connections with like-minded individuals through clans and an interactive chat system.

The Technologies that will be used in our project are:

- HTML
- CSS
- SASS
- React JS
- Node JS
- Express JS
- Mongo DB

Background

In the ever-evolving landscape of technology and coding, the demand for cohesive and engaging platforms that cater to coding enthusiasts continues to grow. Groofy Code emerges within this context, aiming to redefine the online coding experience. The project delves into the intersection of coding challenges, problem-solving, and competitive matches, recognizing the need for a unified space where enthusiasts can explore, compete, and collaborate seamlessly.

Problem Definition

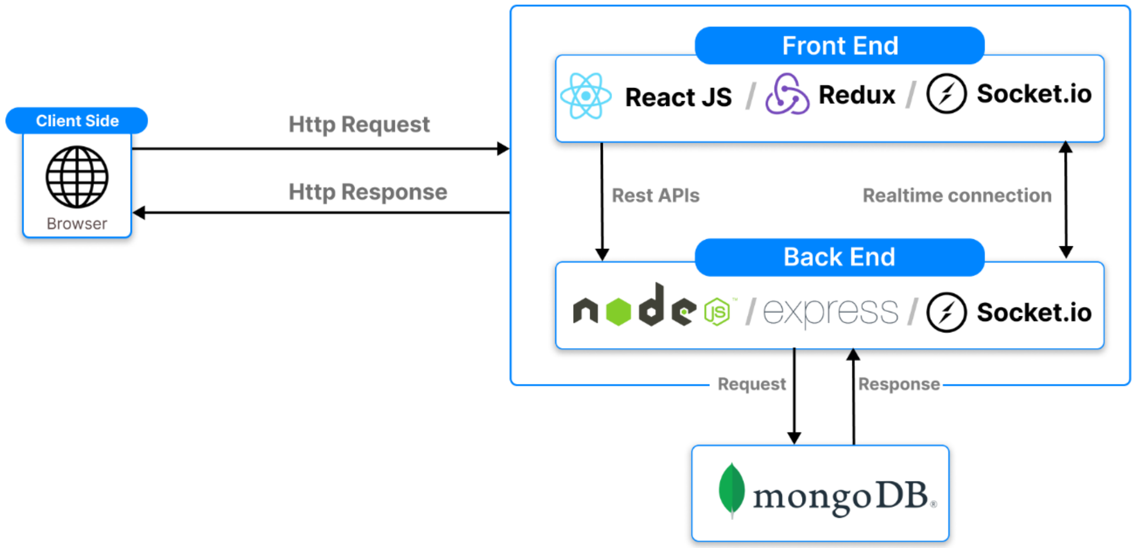
The current landscape of online coding platforms presents a significant challenge as it lacks a cohesive and all-encompassing solution for coding enthusiasts. Existing platforms often specialize in isolated aspects such as coding challenges, competitive matches, or collaborative problem-solving, creating a fragmented experience for users. This fragmentation hinders the development of a unified community where individuals can seamlessly transition between various coding activities, showcase their skills, and engage in social interactions.

The absence of an integrated platform poses challenges for coding enthusiasts seeking a comprehensive and dynamic environment that caters to diverse preferences. Users may find it cumbersome to navigate between different platforms to fulfill their coding interests, leading to a disjointed experience that fails to fully meet their needs.

Our Solution

Groofy Code redefines the online coding experience with a feature-rich platform. The Solver Profile offers a personalized journey, showcasing achievements, progress, and social connections. Problem Solving provides diverse challenges, while 1 vs 1 Challenges add a competitive element with automatic timing and global rating impact. The Clan System fosters collaboration, and Friends and Online Status enhance communication. A dynamic Chat System supports reactions and emojis. Advanced Profanity Detection using machine learning ensures good words. Solvers can customize Unrated Challenges. The Global Rating System evaluates individual and clan performance. Groofy Code's technology stack, featuring React, Express.js, and machine learning, underscores its commitment to a seamless, engaging, and technologically advanced coding environment.

System Architecture



Stakeholders

1. **Developers and Engineers**: These individuals are responsible for creating and maintaining the platform, including its technology stack and features.

2. **Programmers and coders**: Coders are the primary users of the platform, actively participating in problem-solving challenges, 1 vs 1 competition, engaging with community features like clans and friends, and enhancing their skills through solo practice challenges.

3. **Clan Leaders and Members**: Those who lead or participate in clans rely on the platform to foster collaboration, organize challenges, and track clan performance.

4. **Community Managers and Moderators**: These individuals are responsible for maintaining a positive and engaging community environment, enforcing rules, and ensuring that the platform remains welcoming to users.

5. **Business and Marketing Team**: This team is responsible for the growth and promotion of Groofy Code, including attracting new users, retaining existing users, and potentially monetizing the platform through advertisements, subscriptions, or other means.

6. **Investors**: Investors have a financial stake in the success of Groofy Code and are interested in its growth, profitability, and long-term viability.

7. **Technology Partners**: Companies or individuals providing technology solutions or services, such as machine learning tools or hosting services, that are integral to Groofy Code's operation and success.

Functional Requirements

1. User Registration and Authentication

1.1. Users should be able to register with a unique username and password.

1.2. The platform should include email verification for account activation.

1.3. Users must be able to log in securely using their credentials.

1.4. Password recovery/reset functionality should be available.

2. Solver Profile Management

2.1. Users should be able to create and edit their solver profiles.

2.2. Profile information should include a display name, avatar, and optional bio.

2.3. Users can link their solver profiles to their social media accounts.

3. Global Score and Badges

3.1. The platform should calculate and display a global score for each solver.

3.2. Badges should be awarded based on achievements and milestones.

4. Problem Solving

4.1. Solvers can access a diverse range of coding challenges.

4.2. Each challenge should specify the allowed programming languages.

4.3. The platform should support the submission and execution of code.

4.4. A submission history and code editor with syntax highlighting should be available.

5. 1 vs 1 Challenge Mechanism

5.1. Solvers can challenge each other in 1 vs 1 matches.

5.2. Ranked Match

5.2.1 Disconnection Handling:

5.2.1.1. If a player disconnects for more than 30 seconds during a ranked match, the system will initiate a temporary ban for the disconnected player.

5.2.2 Duplicated Source Code Detection:

5.2.2.1. Implement a mechanism to detect duplicated source code.

5.2.2.2. If a player is detected with duplicated source code five times, the system will impose a ban on the player's account.

5.2.3 Draw or Time Expired Handling:

5.2.3.1. In case of a draw (0 - 10), negative trophies will be deducted from both players.

5.2.4 Trophy Calculation:

5.2.4.1. Calculate trophies for the winner and the loser based on the number of wrong answers and the time taken to get the first accepted solution.

5.2.5 Request for Extra Time:

5.2.5.1. Players can send a request for an extra 30 minutes to their opponent during a match.

5.2.5.2. The opponent must accept the request for the additional time to be added to the match.

5.2.5.3. A limit on the number of times the extra time request can be made should be set.

5.3 Custom Match

5.3.1. Player Leave Option:

5.3.1.1. Players can choose to leave a custom match.

5.3.2. Request for Extra Time in Custom Match:

5.3.2.1. Players can send requests for an extra 30 minutes multiple times during a custom match.

5.3.2.2. The opponent must accept the request for the additional time to be added to the match.

5.3.2.3. A limit on the number of times the extra time request can be made should be set.

5.3.3. Match Type:

5.3.3.1. Custom matches can be set as Team or Individual.

5.3.3.2. The maximum number of team members in a team match is five.

5.3.4. Match Duration:

5.3.4.1. Custom matches can have a variable duration, with the option to request an additional 30 minutes.

5.3.4.2. The total match duration can be set by the match creator.

5.3.5. Judged Match Options:

5.3.5.1. Custom matches can be judged based on penalties, solving the most problems, or scores.

5.3.6. Problem Configuration:

5.3.6.1. Custom matches can include multiple problems, either chosen or randomized.

5.4. Code Protection:

5.4.1. Implement measures to block copy-pasting during custom matches.

5.5. Live Templates

5.5.1. Provide the option for players to use live templates during coding.

6. Clan System

6.1. Solvers can create, join, and manage clans.

6.2. Each clan should have a logo, achievements, and bio.

6.3. Clans should have a ranking based on collective performance.

7. Friends and Online Status

7.1. Solvers can search for friends and send friend requests.

7.2. The platform should display the online status of friends.

7.3. Users can manage their list of friends.

8. Chat System

8.1. Clan members can communicate through an interactive chat system.

8.2. The chat system should support reactions, replies, and emojis.

8.3. Users can customize notification preferences for chat activity.

9. Rating System

9.1. The platform should maintain a global rating system for individual solvers and clans.

9.2. Ratings should be updated based on performance in rated challenges and challenge difficulty.

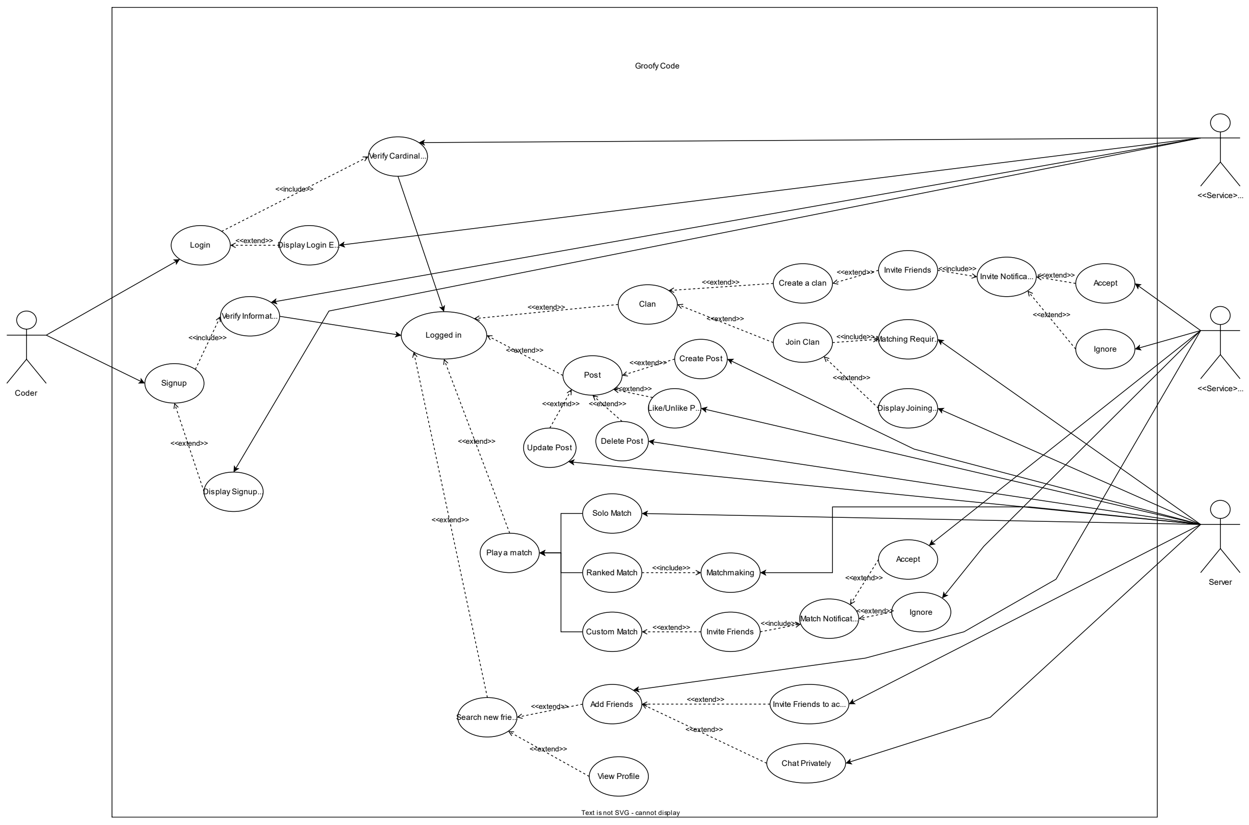
10. Profanity Detector System

10.1. Implement a profanity detector system to monitor and filter inappropriate content in user-generated text.

Non-Functional Requirements:

Non-Functional Requirement name	Definition	Use in project
Scalability	Assesses the highest workloads under which the system will still meet the performance requirements.	The system should scale horizontally to accommodate an increasing number of users.
Performance	Defines how fast a software system (response time) or a particular piece of it responds to certain users' actions under a certain workload.	Response time for key actions, such as code submission and challenge initiation, should be within 2 seconds.
Reliability	Is the probability of failure of software operation for a specified period in a specified environment.	The platform should reliably track and update user scores and ratings in real-time.
Usability	Is the degree to which software can be used by specified consumers to achieve quantified objectives with effectiveness, efficiency, and satisfaction in a quantified context of use.	The user interface should be intuitive, with clear navigation and a responsive design for various devices and screen sizes. Aim for an average user satisfaction rating of at least 4 out of 5 in user feedback surveys.
Portability	Determines how a system or its element can be launched within one environment or another.	The platform should be accessible and fully functional across major web browsers, including Chrome, Firefox, Safari, and Edge.
Security	Assuring all data inside the system or its part will be protected against malware attacks or unauthorized access.	Utilize secure coding practices to protect user data and prevent unauthorized access. Implement encryption for data transmission and storage, including user passwords.

Use Case Diagram:


Groofy Code
Coder
<<Service>...
<<Service>...
Server
Verify Cardinal...
<<include>>
Login
<<extend>>
Display Login E...
Verify Informat...
<<include>>
Signup
<<extend>>
Display Signup...
Logged in
<<extend>>
Clan
<<extend>>
Create a clan
<<extend>>
Invite Friends
<<include>>
Invite Notifica...
<<extend>>
Accept
<<extend>>
Ignore
<<extend>>
Join Clan
<<include>>
Matching Requir...
<<extend>>
Display Joining...
<<extend>>
Post
<<extend>>
Create Post
<<extend>>
Like/Unlike P...
<<extend>>
Update Post
<<extend>>
Delete Post
<<extend>>
Play a match
Solo Match
Ranked Match
<<include>>
Matchmaking
Custom Match
<<extend>>
Invite Friends
<<include>>
Match Notificat...
<<extend>>
Accept
<<extend>>
Ignore
<<extend>>
Search new frie...
<<extend>>
Add Friends
<<extend>>
Invite Friends to ac...
<<extend>>
Chat Privately
<<extend>>
View Profile
Text is not SVG - cannot display