

Peer Learning Platform

Shobhit Kumar (07114803119) & Aayush Sagar (10214803119)

Under supervision of: 1. Dr. ML Sharma (H.O.D)

2. Dr. KC Tripathi

Information Technology – Maharaja Agrasen Institute of Technology

Abstract - The Peer Learning Platform is a web-based tool that enables users to learn and code together in real-time through group video calls and a shared code editor. It is built using technologies such as HTML, CSS, JavaScript, Node.js, Express, Socket.io, Jdoodle, and WebRTC, and supports a variety of programming languages.

The Peer Learning Platform has several potential benefits, including increased collaboration, convenience, support for a variety of programming languages, and real-time feedback. It also has some potential limitations, such as dependence on internet connection, limitations of group video calls, potential for distractions, and security concerns.

Overall, the Peer Learning Platform has the potential to provide a valuable learning experience for users, particularly for those who prefer to learn and code together in a collaborative environment. However, it is important to consider the potential limitations and security concerns when using the platform, and to take necessary precautions to ensure a smooth and secure learning experience.

Key Words: Peer Learning Platform, code, collaboration, HTML, CSS, JAVASCRIPT

1. INTRODUCTION

The Peer Learning Platform is a web-based application that enables users to code and learn together in real-time. It provides a collaborative environment for coding and learning, leveraging technologies such as group video call, chat function, and shared code editor to facilitate real-time code collaboration and communication among users.

Online education has become increasingly popular in recent years, with more and more learners seeking opportunities to learn new skills and advance their careers from the comfort of their own homes. However, online education can be challenging due to various factors such as time management, motivation, communication, and technical issues. The Peer Learning Platform aims to address these challenges by providing a tailored solution for coding and learning that is specifically designed for real-time communication and collaboration.

Using the Peer Learning Platform, users can participate in group video calls powered by WebRTC, chat with each other in real-time, and collaborate on code using a shared code

editor powered by Jdoodle. The platform is built using HTML, CSS, JavaScript, Node.js, and Express, and supports a wide range of programming languages.

The main objectives of the Peer Learning Platform are to provide a collaborative environment for coding and learning, to enable real-time communication and collaboration among users, and to facilitate the sharing and editing of code in real-time.

IMPLEMENTATION OF PROPOSED METHOD/MODEL/ALGORITHM

1.1. Purpose

The purpose of this document is to specify the functional and non-functional requirements for the Peer Learning Platform, a web-based application that enables users to code and learn together in real-time.

1.2. Scope

The scope of this document includes the design and implementation of the Peer Learning Platform, including the user interface, the data flow, the use cases, and the algorithms for setting up a group video call and executing code. It also includes a comprehensive evaluation of the platform, including a detailed analysis of the benefits and challenges of online education and a discussion of the future scope of the project.

1.3. Definitions, Acronyms, and Abbreviations

HTML: Hypertext Markup Language

CSS: Cascading Style Sheets

JavaScript: A programming language used to create interactive web applications

Node.js: A JavaScript runtime built on Chrome's V8 JavaScript engine

Express: A web application framework for Node.js

Socket.io: A JavaScript library for real-time web applications

Jdoodle: A cloud-based code execution platform

WebRTC: Web Real-Time Communication, a set of standards for real-time communication on the web

2. The Peer Learning Platform has the following functions

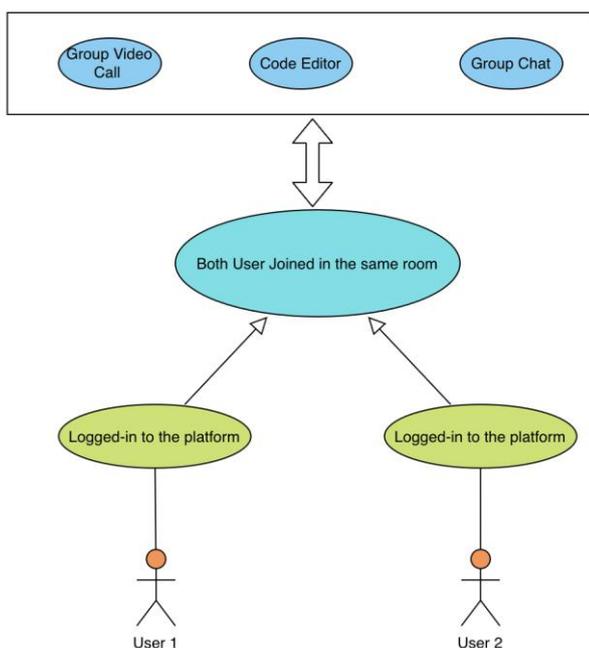
Group video call: Enables users to join a group video call and communicate with each other in real-time. The group video call function uses WebRTC to establish a peer-to-peer connection between users, and supports features such as screen sharing, audio and video streaming, and chat.

Chat function: Enables users to send and receive messages in real-time. The chat function uses Socket.io to establish a real-time connection between users, and supports features such as emojis, file sharing, and notifications.

Shared code editor: Enables users to collaborate on code in real-time. The shared code editor uses Jdoodle to provide a synchronized and collaborative editing.

Code execution: Enables users to execute code in real-time using Jdoodle. The code execution function uses the Jdoodle API to compile and execute code in a variety of programming languages, and provides real-time feedback on the output and any errors.

2.1 User Case Diagram



The use case diagram includes following main use cases:

Create a group: This use case represents the interaction between the user and the system when the user creates a new group and invites other users to join.

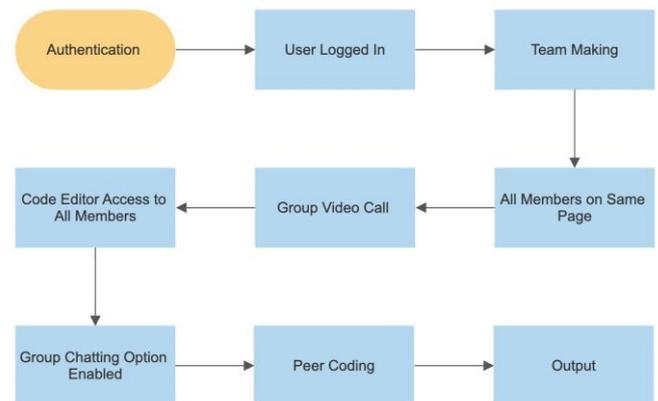
Join a group: This use case represents the interaction between the user and the system when the user joins an existing group.

Start a group video call: This use case represents the interaction between the user and the system when the user starts a group video call and invites other users to join.

Chat with other users: This use case represents the interaction between the user and the system when the user sends and receives messages with other users in real-time.

Code in real-time : This use case represents the interaction between the user and the system when the user code and output is displayed.

2.2 Work-Flow



Algorithm

In the context of the Peer Learning Platform, an algorithm could be used to facilitate the real-time code collaboration and communication among users.

- 1) Set up a group video call using WebRTC.
- 2) Initialize the code editor using Jdoodle.
- 3) Listen for user input in the code editor.
- 4) When a user makes an edit to the code, broadcast the edit to all other users in real-time.
- 5) When a user executes the code, use the Jdoodle API to compile and execute the code in the selected programming language.
- 6) Return the output and any errors to the user in real-time.

This algorithm outlines the steps required to facilitate real-time code collaboration and communication among users on the Peer Learning Platform. It starts by setting up a group video call using WebRTC, which enables users to communicate with each other in real-time. It then initializes

the shared code editor using Jdoodle, which allows users to collaborate on code in real-time. The algorithm then listens for user input in the shared code editor, and when a user makes an edit to the code, it broadcasts the edit to all other users in real-time. When a user executes the code, the algorithm uses the Jdoodle API to compile and execute the code in the selected programming language, and returns the output and any errors to the user in real-time.

3. CONCLUSIONS

The Peer Learning Platform is a web-based application that provides a collaborative environment for coding and learning, enabling users to communicate and share ideas with each other in real-time. The platform leverages technologies such as group video call, chat function, and shared code editor to facilitate real-time code collaboration and communication among users.

Through a literature review, we have identified several challenges associated with online education and real-time collaboration, such as time management, motivation, communication, and technical issues. The Peer Learning Platform aims to address these challenges by providing a tailored solution for coding and learning that is specifically designed for real-time communication and collaboration.

REFERENCES

1. HTML, CSS, and JavaScript:
W3Schools. (n.d.). HTML. Retrieved from <https://www.w3schools.com/html/>
W3Schools. (n.d.). CSS. Retrieved from <https://www.w3schools.com/css/>
W3Schools. (n.d.). JavaScript. Retrieved from <https://www.w3schools.com/js/>
2. Node.js and Express:
Node.js. (n.d.). Node.js. Retrieved from <https://nodejs.org/en/>
Express. (n.d.). Express. Retrieved from <https://expressjs.com/>
3. Socket.io:
Socket.io. (n.d.). Socket.io. Retrieved from <https://socket.io/>
4. Jdoodle:
Jdoodle. (n.d.). Jdoodle. Retrieved from <https://www.jdoodle.com/>
5. WebRTC:
WebRTC. (n.d.). WebRTC. Retrieved from <https://webrtc.org/>