

# MHY PROJECT

**PROJECT REPORT**

MAY 2024

**GROUP MEMBERS:**

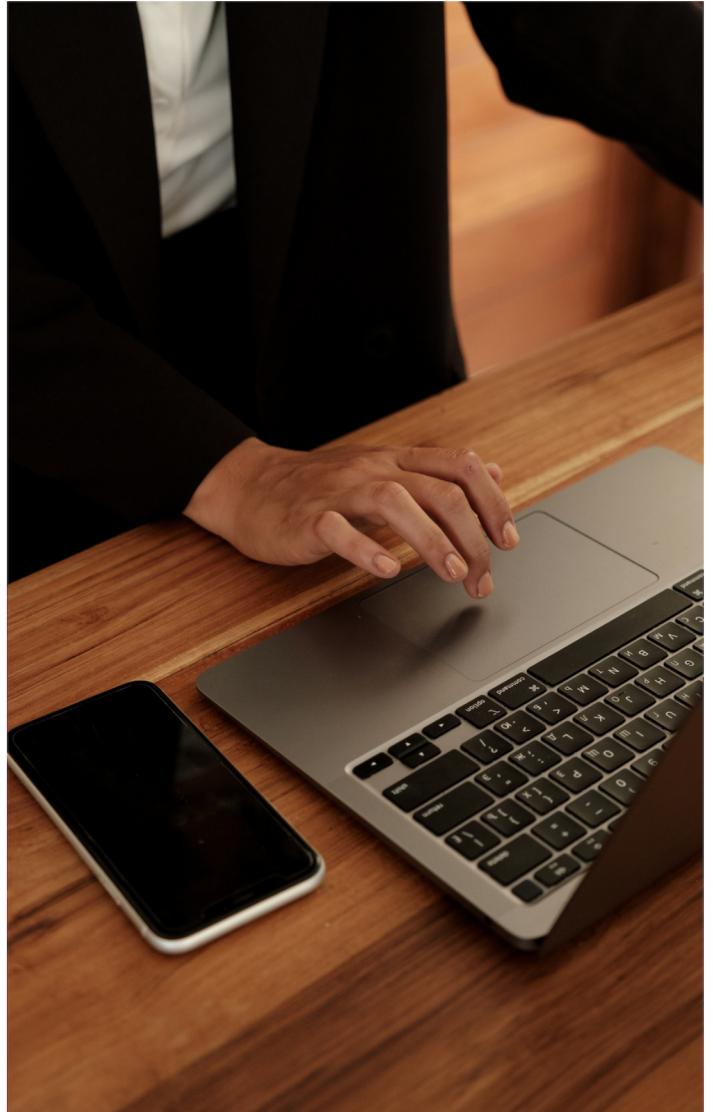
- Yehya Hayati | k21-3309
- Hasan Iqbal | k21-3297
- Mahad Munir | k21-3388

# About

---

**Here is a little description of our project**

Our Computer Networks' project leverages socket programming as its core functionality to facilitate real-time translation from English to French over a network. Using socket programming techniques, we establish a robust communication framework between client and server processes, ensuring seamless data transmission. The integration of machine translation algorithms further enhances our system's functionality, enabling text input in English to be processed and accurately translated into French using advanced language models. Through meticulous implementation and optimization of socket programming, our project showcases the potential for efficient and accurate language translation systems in networked environments.



---

# MORE ON SOCKET PROGRAMMING

Our project harnesses the capabilities of Socket.IO to craft a dynamic chat application enriched with advanced features. Leveraging Socket.IO's event-driven architecture, we seamlessly manage multiple users within a shared chatroom environment. Upon connection, users are directed to specific chatrooms using the 'joinRoom' event, fostering cohesive communication among participants. The 'chatMessage' event facilitates real-time message exchanges, enhancing collaboration and interaction. Moreover, our application elevates user engagement by incorporating file sharing functionalities. Through the 'fileUpload' event, users can effortlessly share files within the chatroom, fostering efficient collaboration and content sharing. This integration of Socket.IO empowers our chat application with robust real-time communication capabilities, creating a vibrant and interactive platform for users to connect and collaborate effectively.

## WHERE WE'RE DOING

Our project features a comprehensive tech stack for optimal functionality and performance. Next.js and React.js power our frontend, ensuring speed and interactivity. Express.js and FastAPI drive our backend, delivering robust APIs and high-speed processing. TensorFlow handles model training efficiently, while Socket.IO enables real-time communication, making our application responsive and dynamic.

# COMPLETE OVERVIEW

## FRONTEND

We've used Next.js and React.js for a fast, SEO-friendly frontend. Next.js handles server-side rendering and routing, while React.js creates interactive UI components with efficient state management. This combination enables us to implement advanced features like real-time updates and dynamic data rendering, ensuring a seamless user experience with optimal performance.

## BACKEND

Express.js and FastAPI are efficient backend frameworks. Express.js, based on Node.js, is popular for creating RESTful APIs and handling HTTP requests. FastAPI, built on Python, prioritizes speed and ease of use, making it ideal for high-performance APIs. Both frameworks offer reliable tools and features, catering to diverse developer preferences and requirements.

## SOCKET PROGRAMMING

Socket.IO simplifies socket programming in web apps by providing a seamless abstraction over WebSockets. It enables real-time bidirectional communication between clients and servers, making it ideal for chat apps and live collaboration tools. With fallback mechanisms for compatibility, Socket.IO streamlines the complexities of socket programming for interactive web development.

## MODEL TRAINING

For our project, we utilize TensorFlow's Keras API for streamlined model training. The efficient training process is enhanced by features like automatic differentiation and model checkpoints, optimizing our workflows for optimal performance.



# CHALLENGES AND SOLUTIONS

## CHALLENGES

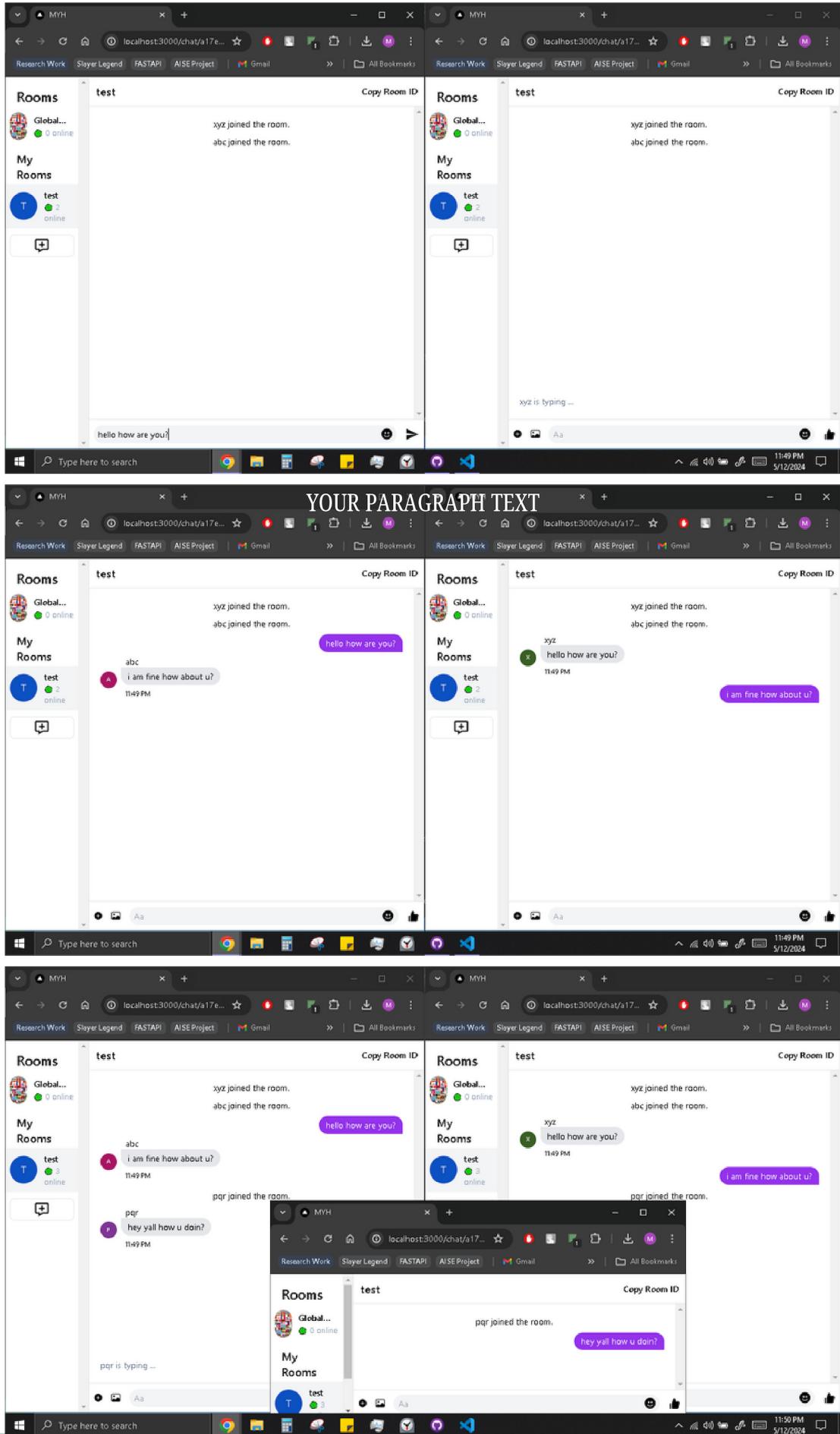
In our project, we faced challenges with socket programming, especially in managing multiple sockets, sending files, and ensuring correct client socket context. Coordinating multiple sockets required careful synchronization to avoid conflicts. Transmitting files over sockets involved managing file sizes and ensuring reliable transfers. Verifying clients' connections to the correct socket context required robust authentication and validation mechanisms. Overcoming these challenges involved thorough testing and implementation of secure socket programming practices for seamless communication.

## SOLUTION

To overcome challenges in socket programming, we implemented strategic solutions. For file transfers, we integrated HTML event listeners, optimizing uploads and ensuring reliable transmissions with error handling. Managing multiple sockets was streamlined using Socket.IO's built-in features like namespaces and rooms, simplifying connection handling. Additionally, we developed a custom socket provider to securely link clients to the correct socket context, enhancing authentication and authorization mechanisms for robust and accurate connections. These measures significantly improved the stability, performance, and security of our socket programming implementation, leading to a smoother and more reliable user experience in our application.



# SNAPSHOTS



# MEET THE TEAM

PEOPLE BEHIND THE PROJECT



**YEHYA HAYATI**



**HASAN IQBAL**



**MAHAD MUNIR**

Our team comprises three skilled members, each bringing valuable expertise to the table. "Yehya Hayati" leads our backend and socket programming efforts, ensuring robust server-side functionality and seamless real-time communication. "Hasan Iqbal" specializes in machine learning, driving our AI and model training initiatives with precision and innovation. Meanwhile, "Mahad Munir" spearheads frontend development, crafting intuitive and visually appealing user interfaces for our application. Together, our diverse skill sets and collaborative spirit enable us to deliver a comprehensive and high-quality product to our users.