Introduction

- A modern **web-based community chat application**
- Built with **React (frontend)** and **Node.js (backend)**
- Uses **Socket.IO** for **real-time communication**
- Provides **two interaction modes**:
  - **Community Mode** – chat with all connected users
  - **AI Mode** – chat directly with AI using OpenAI API

Project Name: Chitchat

Semester project presentation for iteration 0, Team 1

Created by Masih Vahida (Team Leader)

Course: CS673 Fall 2025

System Architecture

- **Frontend (React Web App)**
  - User interface for chat messages
  - Sends and receives messages in real time
  - Option to switch between **Community Chat** and **AI Chat**
- **Backend (Node.js + Socket.IO)**
  - Manages connected clients
  - Broadcasts messages to all users in community mode
  - Routes AI-specific messages to OpenAI API
- **AI Integration (OpenAI API)**
  - Receives user prompts from backend
  - Generates responses in real-time
  - Returns AI reply to the requesting user only

Project Name: Chitchat

Semester project presentation for iteration 0, Team 1

Created by Masih Vahida (Team Leader)

Course: CS673 Fall 2025

Communication Flow

- **Community Mode**
  - User types a message
  - Message sent via Socket.IO to backend
  - Backend broadcasts it to all connected clients
  - Everyone sees the update instantly

- **AI Mode**
  - User types a message
  - Message sent to backend with “AI mode” flag
  - Backend calls **OpenAI API**
  - AI response returned only to that user

Project Name: Chitchat

Semester project presentation for iteration 0, Team 1

Created by Masih Vahida (Team Leader)

Course: CS673 Fall 2025

Benefits

- **Community engagement –** users connect instantly
- **Simplicity –** no complex setup, just open and chat
- **AI-powered assistance –** enriches user experience
- **Extensibility –** easy to add moderation, private rooms, or analytics

Project Name: Chitchat

Semester project presentation for iteration 0, Team 1

Created by Masih Vahida (Team Leader)

Course: CS673 Fall 2025