**Name:** Mohammad Shamil Khan

**Roll Number:** 04072113020

**Course:** Web Application Development

**Project Title:**

No Code Discord Bot Development

**Project Name:**

Its name is Congo. This name is written because Congo are specific specie of parrots that are excellent in talking and mimicry. As this bot also talk like a human with a trained data same like a Congo parrot can talk on which they are trained by humans.

**Mission Statement:**

Our mission is to democratize chatbot creation, making it accessible to all. We empower users without coding skills to effortlessly design human-like conversational experiences and seamlessly integrate AI chatbots into Discord servers, providing affordable and user-friendly solutions.

**Project Motivation:**

The project is driven by the vision to democratize the landscape of chatbot development, eliminating barriers for individuals without coding skills. We are motivated to empower users, enabling them to independently create and customize human-like chatbots. By providing a cost-effective solution, we aim to disrupt the traditional model of expensive developer services, making advanced conversational AI accessible to a wider audience. Our motivation lies in fostering innovation, affordability, and ease of use, ultimately transforming how people engage with and benefit from chatbot technology.

**Thinking Theory:**

**Who:**

There are total 150 Million Active users of discord. Most of them use discord for their gaming, communities, educational channels and so on. When they have a large number of communities they face difficulty to respond to every person and respond to every query so to solve this solution Non-technical users can make the bots for their discord without coding skills and can easily train them on their customize data purely in English.

**What:**

It will break the gap for those who are not economically stable and not able to hire expensive developers to make the bots for the discord so they can easily buy a minimal subscription and can automate their work flow.

**Where:**

This idea will be successful in discord communities, gaming channels on discord

**When:**

It will take almost 4 months to complete.

**Why:**

We are working on this idea because there are very finite number of website providing this service may be 1 or 2 so If we enter this market we can compete easily

**How:**

This idea will be done through the integration of Open-Ai and Discord.js in which the discord server will be integrated with Open-ai and the user will put the data according to his requirement and that data will be stored in MySQL database and then bot uses that data and then manipulate the answer with help of open-ai and then talk like a human.

**Huh:**

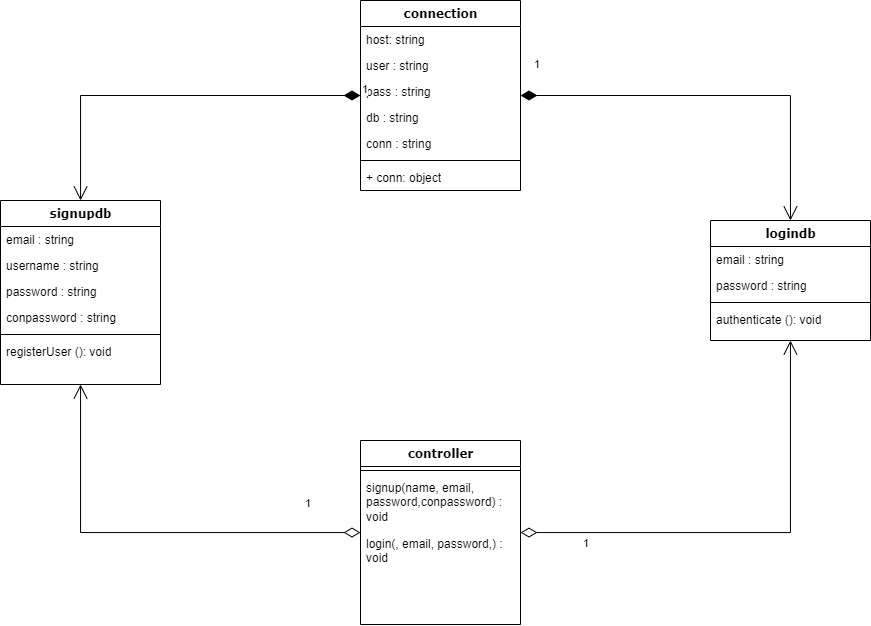
This idea totally make sense as there are total 150 million active users of discord and there are 1 million discord bots and even more discord bots are required every day.

**Audience:**

Through this software we will target gamers, communities owners, admins, educational communities.

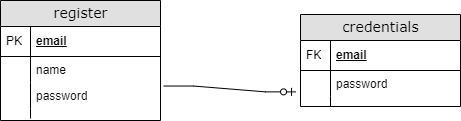
**Class Diagram:**

**Controller Class**The Controller Class serves as the intermediary between the user interface and the backend functionality. It receives data from the interface and delegates it to appropriate classes and files for further processing.**SignupDB Class**The SignupDB Class is responsible for handling user sign-up functionalities. It interacts with the database to store user data securely. Upon receiving data from the Controller Class, it utilizes the connection established in connection.php to store the data in the MySQL database through PHPMyAdmin.**LoginDB Class**The LoginDB Class manages user authentication processes. It verifies user credentials against the stored data in the database. Similar to the SignupDB Class, it relies on connection.php for database connectivity.**connection.php**The connection.php file facilitates the connection between MySQL and PHPMyAdmin. It contains configurations and functions necessary to establish and maintain a secure connection to the database.**Data Flow**Data is received from the user interface by the Controller Class.The Controller Class delegates the data to the appropriate classes such as SignupDB or LoginDB.SignupDB and LoginDB interact with the database using connection.php to store or retrieve user data.Users interact with the website's login page, utilizing the stored data in the database for authentication.

****

**Entity Relationship Diagram:**

**Database Name: Congo**The Congo database consists of two primary tables: register and credentials, each serving distinct purposes in the website's functionality.**Table: register**The register table stores user registration information, comprising three essential columns:name: This column stores the name of the registered users.email: The email column stores the unique email addresses associated with each registered user.password: Passwords chosen by users during registration are securely stored in this column.Table: credentialsThe credentials table is dedicated to storing user authentication data and comprises two crucial columns:email: Similar to the register table, this column stores the email addresses of users.password: User passwords, encrypted or hashed for security purposes, are stored in this column.**Data Organization**Register Table: Stores user registration details including names, email addresses, and passwords.Credentials Table: Solely stores email addresses and corresponding passwords for authentication purposes.**Importance of Tables**Register Table: Facilitates user registration process by storing necessary information for user accounts.Credentials Table: Essential for user authentication, allowing the system to verify user identity during login processes.



**Object-Oriented Programming (OOP):**

Object-Oriented Programming (OOP) is a fundamental paradigm in modern software development, allowing for structured, modular, and reusable code. In the context of the provided codebase, OOP principles are utilized to enhance code organization, maintainability, and scalability.**Controller Class**The Controller Class serves as the central hub for handling user requests and directing them to appropriate components of the backend. It encapsulates the logic for processing incoming data from the user interface and orchestrating interactions between different parts of the system.**SignupDB Class**The SignupDB Class embodies the concept of encapsulation by encapsulating the functionality related to user registration processes. It abstracts away the details of database interactions, providing a clean interface for the Controller Class to interact with.**LoginDB Class**Similar to the SignupDB Class, the LoginDB Class encapsulates the logic for user authentication processes. It shields the rest of the system from the complexities of database querying and authentication algorithms, promoting code reusability and maintainability.**Database Interaction**OOP facilitates a clear separation of concerns between different components of the system. The Controller Class interacts with the SignupDB and LoginDB Classes through well-defined interfaces, allowing for easy substitution or extension of functionality without impacting other parts of the system.**Modularity and Extensibility**By adhering to OOP principles, the codebase is structured in a modular fashion, with each class responsible for a specific set of tasks. This modular design promotes code reusability and extensibility, making it easier to add new features or modify existing ones without introducing unintended side effects.

**Github Link:**

<https://github.com/mshamilkhan/congovanilla>

**URL:**

Congos.ai

**Figma File:**

<https://www.figma.com/file/IKjh3KAb9yvFsTIWtZy9Wt/Untitled?type=design&node-id=0%3A1&mode=design&t=GE5TdNuzIByKSuph-1>