

PROJECT  
SONA



JAVA INSTITUTE FOR ADVANCED TECHNOLOGY

# HACKATHON 2023

**Selection Round**

## Q1. Community Support Platform

### Background:

With the rise of remote work and increasing societal changes, many individuals find themselves needing assistance in various areas, ranging from finding local grocery delivery options to getting mental health support. This challenge aims to create a community support platform where users can post needs, offer help, or find resources in their local communities.

### Task:

Develop a comprehensive web application using PHP that allows:

1. User Authentication:
  - Register and log in.
  - Profile customization, including uploading a profile picture, setting a username, and adding a short bio.
2. Posts:
  - Create, edit, and delete posts.
  - Include an option for users to specify if they are offering help or in need of assistance.
  - Posts should be sortable based on categories like "Groceries", "Medical", "Emotional Support", "Technical Help", etc.
  - Posts should have tags for easier searching.
3. Search and Filters:
  - Search for posts based on keywords, tags, or categories.
  - Filter results based on the user's current location or by date.
4. Interactivity:
  - Users can "upvote" or "downvote" posts to bring attention to the most pressing needs or valuable offers.
  - Commenting system for users to ask questions or coordinate help.
5. Notifications:
  - Users should be notified when someone comments on their post or if a post in their chosen category is made.
6. Responsive Design:
  - The platform should be usable and intuitive on both desktop and mobile devices.
7. Extra Features (Bonus Points):
  - Implement a direct messaging system between users.
  - Allow users to mark posts as "resolved" or "help provided".
  - Incorporate a map view to visualize where help is needed/offered in relation to the user's location.
  - Gamify the platform by giving points to users who help others, creating a leaderboard.

## **Q2. AI-Powered Customer Support Chatbot**

### **Background:**

The goal is to develop a web-based chatbot system using PHP, a relational database (e.g., MySQL), and natural language processing (NLP) libraries to provide efficient and effective customer support to users. This scenario will test your team's PHP development skills, database management, and chatbot implementation abilities.

### **Challenge:**

Your team's task is to build an AI-powered customer support chatbot system with the following functionalities:

1. **User Registration and Authentication:** Implement user registration and login functionalities to allow users to create accounts and securely access their profiles.
2. **Chatbot Interface:** Create a chatbot interface that allows users to interact with the chatbot. Users should be able to ask questions, seek assistance, or report issues.
3. **Database Integration:** Use a database to store user profiles, chat histories, and frequently asked questions (FAQs).
4. **NLP Integration:** Integrate natural language processing (NLP) libraries or APIs (e.g., NLTK, spaCy) to enable the chatbot to understand and respond to user queries intelligently.
5. **User Profiles:** Allow users to provide and update their personal information through the chatbot interface.
6. **FAQ Integration:** Include a feature where users can access frequently asked questions and receive relevant answers from the chatbot.
7. **Real-Time Chat:** Enable real-time chat functionality, ensuring that users receive prompt responses from the chatbot.
8. **Escalation to Human Agents:** Implement a feature that allows the chatbot to escalate complex queries or issues to human customer support agents when necessary.

## **Q3. Real-Time Collaborative Task Management System**

### **Background:**

The challenge is to create a web-based application using PHP and a relational database system (e.g., MySQL) that allows users to collaboratively manage tasks and projects in real-time. This scenario is suitable for teams who want to showcase their PHP and database integration skills.

### **Challenge:**

Your team's task is to build a real-time collaborative task management system with the following functionalities:

1. **User Registration and Authentication:** Implement user registration and login functionalities to allow users to create accounts and securely access their profiles.
2. **Task Creation and Assignment:** Users should be able to create tasks, assign them to team members, and set due dates.
3. **Real-Time Updates:** Implement real-time updates for tasks, so when a task is assigned or updated, all team members can see the changes instantly without refreshing the page. You can use technologies like WebSocket or AJAX for this.
4. **Task Status Tracking:** Users should be able to update the status of tasks (e.g., to-do, in progress, completed).
5. **Project Management:** Organize tasks into projects or categories. Users can create, edit, and delete projects.
6. **Commenting and Messaging:** Allow users to leave comments on tasks for discussion and send direct messages to team members.
7. **Deadline Reminders:** Send email or in-app notifications for upcoming task due dates.
8. **Search and Filtering:** Implement search and filtering options to help users find tasks and projects quickly.
9. **Data Storage and Database Integration:** Ensure that all task and user data is stored securely in a relational database.