Members of the project :

- Kimia Keivanloo
- Sara Khalili
- Amir Hossein Sabri

1. Which of the types of information systems do you consider suitable for your project? (Explain) :

This application can be categorized as a "Content Management System (CMS)" or a "Social Networking System”

- Content Management System (CMS) :

Our project allows users to create and manage profiles, skills, and projects. This is similar to the core functionality of a CMS. Users can input, edit, and display content (profiles and projects) with associated data, such as images, text, and links. The "Project" model, in particular, resembles content items, and the "Tag" model could be used for categorizing content. You can consider adding features like rich text editing for more CMS-like functionality.

- Social Networking System :

Our application also exhibits features commonly found in social networking platforms. Users can create profiles, send and receive messages, and review projects. The "Profile" model serves as user profiles, the "Message" model allows for private messaging between users, and the "Review" model supports user interactions and voting on projects. You might consider expanding social features, such as user connections, comments, and notifications, to make it more like a social networking system.

2. Who are the stakeholders of the system? Explain in detail :

The stakeholders of the system for your project can be categorized as follows:

Users: These are the primary stakeholders who interact with your system. Users create and manage personal profiles, post projects, communicate through messaging, and provide reviews and votes. They are central to the platform's operation and success.

Project Owners: Users who create and manage their projects on the platform are also stakeholders. They benefit from the visibility and engagement your system provides for their projects.

Administrators: Individuals responsible for managing and maintaining the system, ensuring its functionality, and moderating content. They play a crucial role in system governance.

Developers: Those responsible for the system's development, maintenance, and improvement are also stakeholders. They contribute to the system's technical aspects.

Community or User Base: The broader community or user base who engage with the platform's content, such as reviewing and voting on projects, are also stakeholders. They contribute to the platform's dynamic nature.

3. As a system analyst, answer the following questions:

A) What problems has this system solved for end-users or what opportunities has it created?

Improved Networking and Collaboration: The system has allowed users to create and manage their profiles, making it easier for them to connect with others who share similar interests or skills. This enhances networking and collaboration opportunities among users.

Efficient Communication: The messaging system in the platform has improved communication between users. It enables them to send messages, exchange information, and discuss projects, facilitating efficient and direct communication.

Project Exposure: Project owners can showcase their work to a broader audience, increasing their project's visibility and chances of getting recognized or gaining collaborators or job opportunities.

User Engagement: The review and voting system provides users with a way to engage with projects and offer feedback. It encourages user participation and interaction, making the platform more engaging and dynamic.

Data-Driven Decision-Making: The system collects data on project reviews and votes, which can be used for informed decision-making. Users can make choices based on the popularity and reception of projects within the community.

Skill Promotion: Users can display their skills and expertise in their profiles, potentially attracting interest from potential employers or collaborators seeking those skills.

Content Categorization: The Tag model allows for the categorization of projects, making it easier for users to discover content of interest.

B) Express the system requirements from the three aspects of knowledge, process, and communications. In your explanation, address the following points as well :

. What entities must exist in the system at minimum of 3 and Maximum of 5 :

Profile: This entity is essential and should be present in the system. It stores information related to end-users, including their names, email addresses, biographies, skills, and social media links.

Project: The Project entity is essential for managing and displaying user-created projects. It should include details such as project titles, descriptions, links to demos and source code, and associated tags.

Review: To capture user feedback and interaction with projects, the Review entity is necessary. It stores information about user reviews, which is a fundamental aspect of your system.

Tag: The Tag entity is important for categorizing and labeling projects, helping users discover content based on their interests or categories.

Message: For facilitating communication between users, the Message entity is vital. It stores message content, sender and recipient information, and message timestamps, enabling user-to-user interactions.

. What roles exist in your system? At least 2 and Maximum 3 :

User Role: This is the primary role in your system. Users are the individuals who interact with the platform. They create and manage profiles, post projects, communicate through messaging, provide reviews and votes, and engage with content. The User role is central to the system's operation.

Administrator Role: Administrators play a key role in managing and maintaining the system. They are responsible for system governance, ensuring its functionality, moderating content, and handling any administrative tasks related to user accounts or content management.

Guest Role (Optional): While not mandatory, you can introduce a "Guest" role for individuals who visit the platform without creating user accounts. Guests may have limited access, such as viewing public content but not engaging in messaging or reviews.

. What tasks must the above minimum roles perform in the system :

User Role:

Create and Manage Profiles: Users should be able to create their profiles, providing information such as names, email addresses, biographies, skills, and links to social media profiles. They should also have the ability to edit and update their profiles as needed.

Post Projects: Users need the capability to create new projects by providing project titles, descriptions, links to demos and source code, and associating tags with their projects.

Engage with Content: Users must be able to engage with content by providing reviews and votes on projects created by themselves or others.
Use Messaging: Users should be able to send and receive messages, facilitating communication with other users.

Administrator :

System Maintenance: Administrators are responsible for maintaining the overall functionality of the system, ensuring that it runs smoothly and addressing any technical issues.

Content Moderation: Administrators should monitor and moderate user-generated content, ensuring it complies with platform guidelines and policies.

User Account Management: Administrators have the authority to manage user accounts, which may involve actions such as suspending or banning accounts in case of violations.

Guest Role:

View Public Content: Guests should be able to browse and view publicly accessible content on the platform, including project details, profiles, and project reviews. They can explore the platform without the need to register.

. What input does the user provide and what output does the user receive for the execution of each of the above tasks?

User Role:

Create and Manage Profiles:

Input: Users provide personal information such as names, email addresses, biographies, skills, and social media links.
Output: Upon successful input, the system updates the user's profile, and the user can view their updated

Post Projects:

Input: Users input project details, including titles, descriptions, links to demos and source code, and associated tags.
Output: After input, the system displays the project to the user and other users on the platform.

Engage with Content (Reviews and Votes):

input: Users provide feedback by submitting reviews and votes on projects.
Output: The system records the user's review and vote, and other users can see their contributions in project reviews and vote counts.

Use Messaging:

Input: Users provide message content, recipient information, and sender information.
Output: The system sends the message to the specified recipient, and the recipient can view and respond to the message.

Administrator Role:

System Maintenance:

Input: Administrators monitor system performance and may input commands for maintenance or system updates.
Output: The system performs maintenance tasks, resolves issues, and provides feedback or reports on the status of system health.

Content Moderation:

Input: Administrators review user-generated content and may take actions like flagging or removing inappropriate content.
Output: The system enforces content moderation decisions, which may involve removing or blocking certain content.

User Account Management:

Input: Administrators manage user accounts, including actions like suspending or banning accounts.
Output: The system applies the requested account management actions, affecting the status of the user accounts accordingly.

4 .As the executive manager of the system, which part of the system do you see as needing the system analysis? (Explain your reason) :

Reason:

The Messaging System is a critical component of the platform that enables users to communicate and collaborate. It plays a vital role in user interactions, facilitating direct messaging between users for project discussions, inquiries, or collaborations. Proper analysis of the Messaging System is important for the following reasons:

Efficient Communication:

System analysis can ensure that the messaging system is designed and optimized for efficient and reliable communication. This includes assessing message delivery, notification mechanisms, and user experience.

Scalability:

As the user base grows, the messaging system needs to handle an increasing volume of messages. System analysis can evaluate the system's scalability to accommodate a larger number of users.

Security and Privacy:

Messaging systems often involve sensitive information. System analysis can address security and privacy concerns to safeguard user data and communication.

User Experience:

Analyzing the messaging system can help improve the user experience by assessing features like message search, message history, and the user interface for ease of use.

Integration:

System analysis can consider integration with other parts of the platform, such as user profiles and project discussions, to ensure a seamless user experience.

5. Provide a reason and a method of communication between two systems that can be used for connecting your system with another system (mention the other system) :

Reason:

OAuth is a commonly used authentication protocol that allows users to grant limited access to their resources (in this case, their user profile information) without sharing their credentials (username and password) with a third-party application or system. Integrating with an OAuth system can provide several benefits for user convenience and security:

Single Sign-On (SSO):

Users can log in to your system using their credentials from a trusted external system, streamlining the login process.

Enhanced Security:

User data remains protected because your system does not store sensitive credentials. OAuth tokens ensure secure communication between the systems.

User Trust:

Users are more likely to trust an authentication process that is handled by a recognized external service.

Method of Communication:
To achieve this, you can implement OAuth as the method of communication between your system and the external authentication system (e.g., using OAuth libraries and frameworks). Here's how it works:

A. **User Initiates Login:**
When a user wants to log in to your system, they are redirected to the external authentication system (e.g., Google, Facebook, or a custom OAuth provider).

B. **User Grants Permissions:**
The user grants permission for your system to access specific user data, such as their name, email, or profile picture.

C. **Authentication Token:**
The external system issues an authentication token to your system, confirming the user's identity.

D. **Access to User Data:**
Your system can use the token to access the user's data from the external system, making it available within your application.

6. What implementation technology is suitable for each of the three aspects of the system?

In the context of my project, a Django web application for managing user profiles and projects, different implementation technologies are suitable for various aspects of the system:

Backend Development:
For the backend development of the system, Django is the most suitable technology. Django is a high-level Python web framework that provides a robust and versatile platform for building web applications. It offers features such as user authentication, database management, and built-in admin panels, making it an excellent choice for implementing the backend of a system that handles user profiles, projects, and interactions.

Frontend Development:
For the frontend development, a combination of HTML, CSS, and JavaScript, along with popular frontend frameworks like React or Vue.js, is recommended. Django can work seamlessly with these technologies through RESTful APIs. HTML and CSS handle the structure and styling of web pages, while JavaScript, along with frontend frameworks, can provide dynamic and interactive features, enhancing the user experience.

Database Management:
For database management, SQLite3 is currently used in your project, and it is suitable for smaller to medium-sized applications. However, if the project grows and requires more scalability and performance, you may consider transitioning to a more robust database management system such as PostgreSQL, which Django supports well. PostgreSQL is suitable for systems that need to store and manage a larger volume of data efficiently.

Overall, Django serves as the core technology for backend development, while a combination of HTML, CSS, JavaScript, and frontend frameworks is recommended for the frontend development. The choice of the database management system depends on the scale and specific requirements of the project, with SQLite3 serving adequately for smaller-scale applications and PostgreSQL being a scalable option for larger systems.