Project Documentation

Project Name: Facebook

Project Description:

Facebook is a social networking website where users can connect with friends and family, share photos and videos, and join groups. It uses microservice architecture in which multiple microservices work together to give seamless experience to users. The backend of the application is built using Spring Framework whereas the frontend is built using Angular/React.

Major Modules:

- User Registration and Login
- Posts management
- Profile Management
- Friend Request Management

Architecture:

This application uses Microservice Architecture to make it a robust, scalable, and easy-to-maintain application. Each microservice is responsible for individual tasks and different microservices communicate with each other using REST APIs. The following microservices will be implemented:

- **User Microservice**: This microservice will be responsible for user authentication and managing user profile. It will provide APIs for user registration, authentication, and updating user profiles.
- **FriendRequest Microservice**: This microservice will be responsible for managing friend requests which includes sending friend requests, accepting/declining friend requests, and viewing the friend request list.
- **Posts Microservice:** This microservice will manage posts which include posting text, photos, and videos and viewing and liking posts shared by others.
- **FriendProfile Microservice:** This microservice will help the user to view a friend's profile, viewing and liking the posts.
- **Gateway Service:**
    - Acts as the entry point for all requests to the microservices.
    - Uses Spring Cloud Gateway to handle routing, load balancing, and traffic management.
    - Uses Spring Cloud Consul for service discovery and registration.

Technology Stack:

- **Backend:** Spring Boot, Spring Data, Spring REST, Spring Cloud Consul
- **Frontend**: Angular/React, Typescript/JavaScript HTML5, CSS3, Bootstrap
- **Database**: MySQL/NOSQL
- **Security** - OAuth/Spring Security

Non-Functional Requirements:

- Performance: The application should be fast and responsive, with quick load times and minimal lag or delay.
- Scalability: The application should be able to handle large amounts of traffic and scale as needed to accommodate growth.
- Security: The application should be secure and protect user data from unauthorized access, with measures such as encryption and secure authentication.
- Reliability: The application should be reliable and available, with minimal downtime or outages.
- Usability: The application should be easy to use and intuitive, with a clear and consistent user interface.
- Maintainability: The application should be easy to maintain and update, with clear and well-organized code that is easy to understand and modify.
- Compatibility: The application should be compatible with a wide range of devices and browsers, with support for different screen sizes and resolutions.

Final Deliverables:

- Application archive (.jar) with source code
- Database DDL Script
- Complete Source code
- Sample screenshots of important screens

User Stories:

US 01: User Registration

As a new user of Facebook, I should be able to create an account with a unique username and password and use this platform to post new updates, view and like updates of my friends and family.

Acceptance Criteria:

- When accessing the application, I should see a prominent option or link to "Register" or "Create an Account."
- Clicking on the "Register" or "Create an Account" option should take me to a registration page.
- On the registration page, there should be clearly labeled input fields for entering the following information:
    - **Username**: Two text fields for entering the username as first name and surname which accepts alphanumeric (letters and numbers). The first letter of each word should be capitalized.
    - **Email Address**: A text field for entering the user's email address. Email Address should be in a valid email format with domains like 'com' or 'org' or 'in'.
    - **Date Of Birth**: It should be a past date. Must be at least 13 years old.
    - **Gender:** it should be either 'Male', 'Female' or 'Others'
    - **Password**: A text field for entering a secure password. Password should contain lowercase, uppercase, digits, and special characters at least one each. The minimum length of the password should be 8 and maximum 16.
    - **Confirm Password**: A text field for re-entering the password to confirm it. "Passwords do not match" alert message should be shown while entering in the confirm password field until passwords do not match.
- After successful registration, the system should redirect me to a "Home" page, indicating that my account has been created.
- The system should enforce uniqueness for usernames, ensuring that no two users can register with the same username.
- The system should provide appropriate feedback if the registration fails due to any technical issues, such as a server error or database connectivity problems.
- The system should provide a "Forgot Password" option to allow users to reset their passwords if they forget them, using a secure password reset mechanism.

US 02: User Login

As a user, I want to be able to log in to my account with unique usernames and passwords so that I can access my profile.

Acceptance criteria:

- Validate the credentials provided by the user.
- If 3 failed attempts are made within 30 seconds, the Customer Microservice circuit should go in the open state for 1 minute. (Implement it using Resilience4j).
- Display a running timer of 1 minute while the circuit is in the open state also display a proper message if the user attempts to log in when the circuit is in the open state.
- Customer should be able to reset the password if he forgets his password linked to his email. Verify his email Id for resetting the password.
- Display proper error messages if the credentials are wrong.
- After successful login, the user is redirected to the home page. (The home page will have the posts shared by people and friends, also a button to navigate to the profile)

US 03: Post Updates:

As a user, I want to be able to post updates (text, photos, and videos) to my profile so that I can share them with my friends and view updates from others.

Acceptance Criteria:

- As a user, I should be able to create a new post by selecting the "Create Post" option.
- The post-creation form should allow me to upload photos and videos from my device or capture them using the device's camera.
- I should be able to add a caption or description to the post, providing context or additional information.
- The post creation form should provide options to select the privacy settings for the post, such as public, friends only, or custom.
- After posting, the content (photos, videos, and text) should be displayed on my profile or in the selected group.

- Other users should be able to view, like, comment, and share my post based on the privacy settings I have chosen.
- The post should be displayed in a visually appealing format, with the media (photos/videos) and text content properly formatted and displayed.

US 04: View and Update Profile

As a user of the Facebook application, I want to be able to view and update my profile information, so that I can keep my profile up-to-date and provide accurate information to my friends and connections.

Acceptance Criteria:

- As a user, I should be able to view my profile information, including my name, profile picture, bio, and other relevant details.
- As a user, I should be able to update my profile information, such as my name, profile picture, bio, contact information, and other relevant details.
- Any changes made to my profile information should be reflected across the application and visible to my friends and connections.
- The profile update functionality should provide appropriate validation and error handling to ensure data integrity and a smooth user experience.
- The profile view and update functionality should be accessible from the user's account settings or profile page, with clear navigation and an intuitive user interface.

US 05: Friend Requests management

As a user of the Facebook application, I want to be able to manage friend requests, so that I can connect with other users and expand my network.

Acceptance Criteria:

- As a user, I should be able to send friend requests to other users by searching for their profiles and selecting the "Add Friend" option.
- As a user, I should be able to accept or decline friend requests.

- If I accept a friend request, the user should be added to my friend's list and vice versa.
- If I decline a friend request, the user should not be added to my friend's list and vice versa.
- As a user, I should be able to view a list of pending friend requests and manage them (accept or decline) from a dedicated section in my profile.
- As a user, I should be able to search for specific users and send them friend requests directly from their profile page.

US 07: view friend's profile
As a user of the Facebook application, I want to be able to view the profile of my friends, so that I can learn more about them and stay updated with their activities and like their posts.

Acceptance Criteria:

- As a user, I should be able to navigate to a friend's profile page from my friend list or search results.
- The friend's profile page should display relevant information such as their profile picture, cover photo, name, bio, location, and any other publicly available details.
- I should be able to see the friend's recent posts, photos, and other shared content on their profile page.
- The friend's profile page should provide options to interact with their content, such as liking.
- The privacy settings of the friend's profile should be respected, and I should only be able to view the information and content that is shared with me based on their privacy settings.
- The profile page should have a user-friendly layout and design, making it easy to navigate and consume the friend's information and content.

US 08: view friend list
As a user of the Facebook application, I want to be able to manage my friends list, so that I can organize and control my connections on the platform.

Acceptance Criteria:

- As a user, I should be able to view my current friend list, displaying the names and profile pictures of my friends.
- I should be able to search for specific friends within my friend's list.
- I should be able to add new friends to my friend's list by sending them friend requests.
- I should be able to remove friends from my friend list, effectively unfriending them.
- I should be able to block or unfollow friends, limiting their access to my profile and updates.
- I should be able to organize my friends list into custom categories or groups for better management.

US 09: view/Like the posts

As a user of the Facebook application, I want to be able to view posts from my friends and connections and like them to show my appreciation or support.

Acceptance Criteria:

- As a user, I should be able to scroll through my news feed or timeline to view posts from my friends and connections.
- Each post should display the content, such as text, photos, or videos, along with the name and profile picture of the person who posted it.
- I should be able to like a post by clicking on a designated button or icon.
- The number of likes on a post should be updated in real-time when a user likes it.
- I should be able to see the list of users who have liked a post.
- The option to unlike a post should be available if I change my mind.
- The privacy settings of the post should be respected, and I should only be able to view and like posts that are shared with me based on the post owner's privacy settings.