Test Progressive Assistants	
Web Design and Programming	
For 23-1 Generation	
Valid on *Short Semester Year 2022/2023*	

1. Peserta TPA tidak diperkenankan untuk:

    a. Mempublikasikan jawaban yang telah dibuat secara sengaja ataupun tidak disengaja.

    b. Membuka dan/atau menyalin jawaban dari buku, video, ataupun peserta lainnya.

    c. Melakukan tindakan yang menyebabkan jawaban dicontek oleh orang lain atau kelompok lain, baik disengaja maupun tidak disengaja.

    d. Melakukan tindakan kecurangan lainnya.

2. Jika peserta TPA dan/atau terbukti melakukan tindakan seperti yang dicantumkan pada butir ke-1, maka nilai TPA yang melakukan kecurangan, baik menyontek atau dicontek, akan dinolkan sesuai dengan peraturan yang berlaku. Dan akan diberikan sanksi sesuai prosedur yang berlaku.

3. Jawaban yang dapat diterima dan dinilai adalah jawaban yang dikumpulkan sebelum batas waktu yang telah ditentukan.

Soal

Case

faREbook

FaREbook is a platform to connect people and allow them to interact with friends, family, and acquaintances online. Users create profiles where they can share personal information, photos, and updates about their lives. They can also add "friends" to their network, forming connections with others on the platform.

The project manager, **RE** wants you to create a website for **FaREbook.** Since the platform is intended for global distribution

Upon development, **RE** expects you to **know what you are using**, and **why you are using it. RE** wants you to use **Git** version control so both you and **RE** can keep track of your work by seeing your commits every day. You could use **GitHub** for the git repository manager. Use the github repository given by *Assistant Development* to make a commit of your projects. **RE** expects you to uphold your **integrity** so make sure every time you're about to present your work to **RE**, you're presenting what you had done until the **last commit before the deadline.**

In this project, you are asked to use **GO** language as the backend language to create **GraphQL API**, your backend must be accessible by only 1 endpoint. You are also required to use **Vite** as a build tool for your **React** frontend framework to develop your web application. The language that you must use for the **React** framework is **TypeScript.**

For the database you must use **PostgreSQL.** You are **free to create and design the database** in any section, but you must have a **strong logical reason why**. There are some factors that you should notice on development such as **performance, security, design, and tools understanding.**

In the performance factor, you are asked to use **Redis** to cache the result of any API data you consume. In the design factor, the web design must be responsive with 3 breakpoints, **desktop, tablet,** and **mobile**.

Since **RE** notices that so many junior developers don't have a strong foundation in **CSS, RE** hopes your **CSS** skills will **increase significantly** after you **finish** this project. With the intention to improve your **CSS skill** you are prohibited to use any **CSS library or UI component libraries** such as Bootstrap, Material UI, Prime, etc. **RE** wants to share the wonderful feeling of using **CSS/SASS/SCSS** styling language, so you are asked to use it.

Other than **UI** and **Design,** please consider the **UX** of the application, because **the experience** is what makes customers loyal to the web, **not the design.**

In the security factor, you are asked to use **JWT Token** for login authentication process. You are also asked to implement **SALT** for user's password to make it more secure against attack. to make your application more secure against attack. If you find any bugs in any technology you are using, report bugs to the official website and write your documentation of what bugs you had found.

In the tools understanding factor, you are asked to just simply not just use the tools, but **understand** *what*, *why*, *how* and anything that *related* to the tools you are using. For example: Do the tools **implement any design pattern or infrastructure pattern** that you could add to your project.

While working on this project, remember what's the main functionality of the web, focus on **learning something new**, find some motivation when you're unmotivated, rest when you are tired, work smartly, be creative, and **have fun**😀.

Here are the **Project Requirements** you need to build for these projects:

➢ **General**

- Must pay attention to performance in the frontend, backend, and database.
- Must be able to explain the purpose and the reason of the code and action.

➢ **All Page**

- Must have two themes: *Light* **and** *Dark*. And for the **default** theme is *Light* theme.
- Must be able to **toggle** between those two themes.
- Must be **responsive** on desktop, tablet, and mobile.
- Must have proper *UX*.
- Must have proper authentication & authorization.
- Must display **Navigation Bar** Component for the **authorized page** for logged-in user.
- Must display **error message** if an error were to occur.
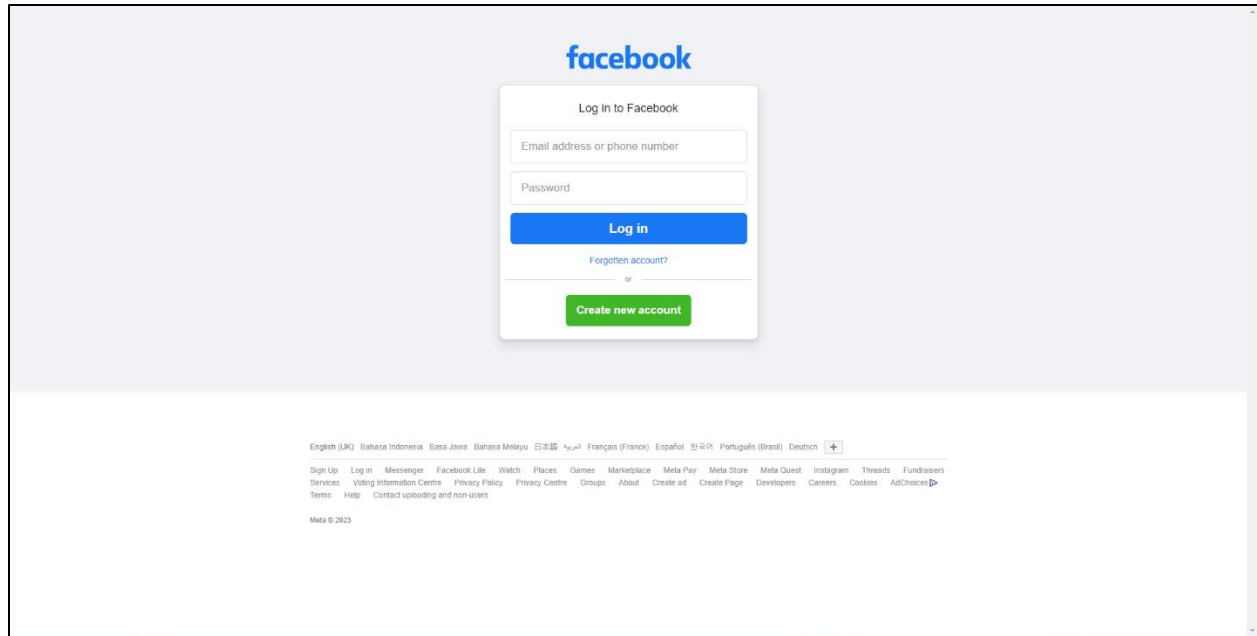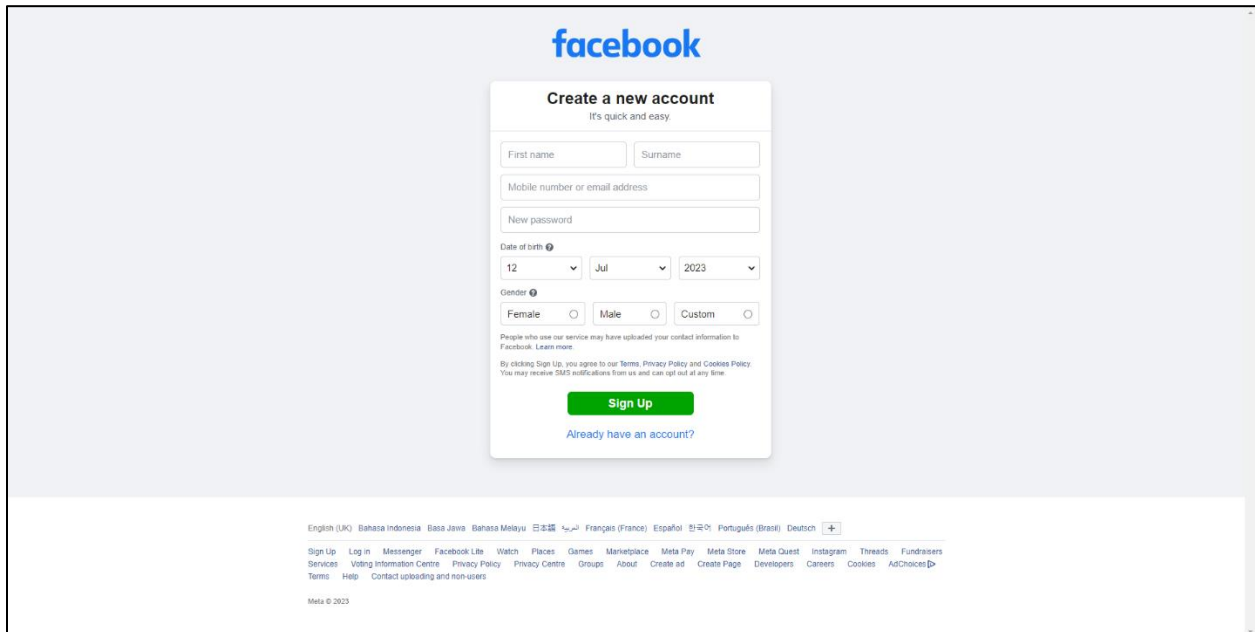- **May** follow the figure of page.

➢ **Login Page**

Figure 1. Login Page

- Must be **authorized** only to guest.

- Must **redirect** here if guest were to enter unauthorized page.

- Must use **email** and **password** to log in.

- Must only allow **activated account** to log in.

- Must contains link to **Register Page** and **Forgotten Account Page**.

➢ **Register Page**

Figure 2. Register Page

- Must be **authorized** only to guest.
- Must register a new **unactivated account** using email and password and user personal information such as first name, last name, email address, password, date of birth, and gender.
- Must include **validation** in the field at least 4 (the validation must be different).
- Must only allow an email address to be used by **one account**.
- Must send **activation link** with random code to email for account activation.
- Must contains link to **Login Page**.

➢ **Account Activation Page**

- Must be **authorized** only to guest and can be accessed only from valid activation link.
- If user is valid, **activate** the user account.
- Must be able to redirect user to **Login Page**.

> **Forgotten Account Page**

Figure 3. Forgotten Account Page

- Must be **authorized** only to guest.

- Must only allow **email address** that is **connected** to an account.

- Must send reset **password link** with random code to email for account reset password.

- Must contains link to **Login Page**.

> **Reset Password Page**

- Must be **authorized** only to guest with valid code.

- Must contains two fields: *Password* and *Repeat Password.*

- Must make sure *Password* and *Repeat Password* is the same.

- Must make sure *Last password* and **the new password** is not the same.

- Must only allow **replacing the password** of user with *Password* after all the requirements are fulfilled. After that redirect user to **Login Page**.
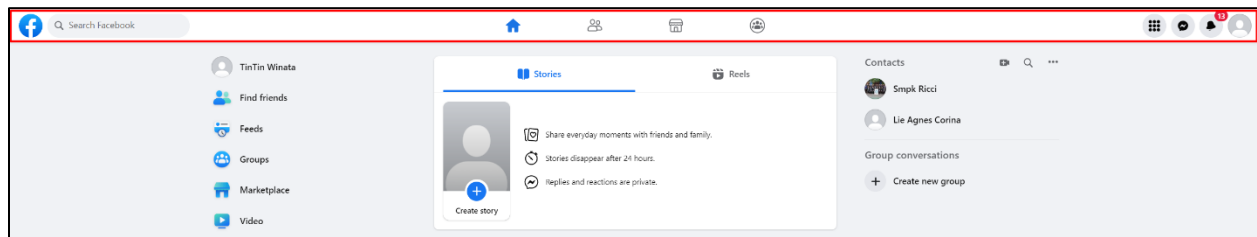
➢ **Navigation Bar Component**

Figure 4. Navigation Bar Component

- Must display logged user profile picture on the right side.

- Must allow searching for posts, groups, marketplaces, and users in **Search Page**.

- Must be able to log out.

- Must display number of notifications.

- Must contains links to **Home Page**, **Friends Page**, **Marketplace Page**, **Group Page, Messenger Page**, **Notification Page**, and **User Profile Page**.

➢ **Footer Component**

- Must display at least 5 useful site links (ex: *privacy*, *terms*, *cookies*, *etc.*)
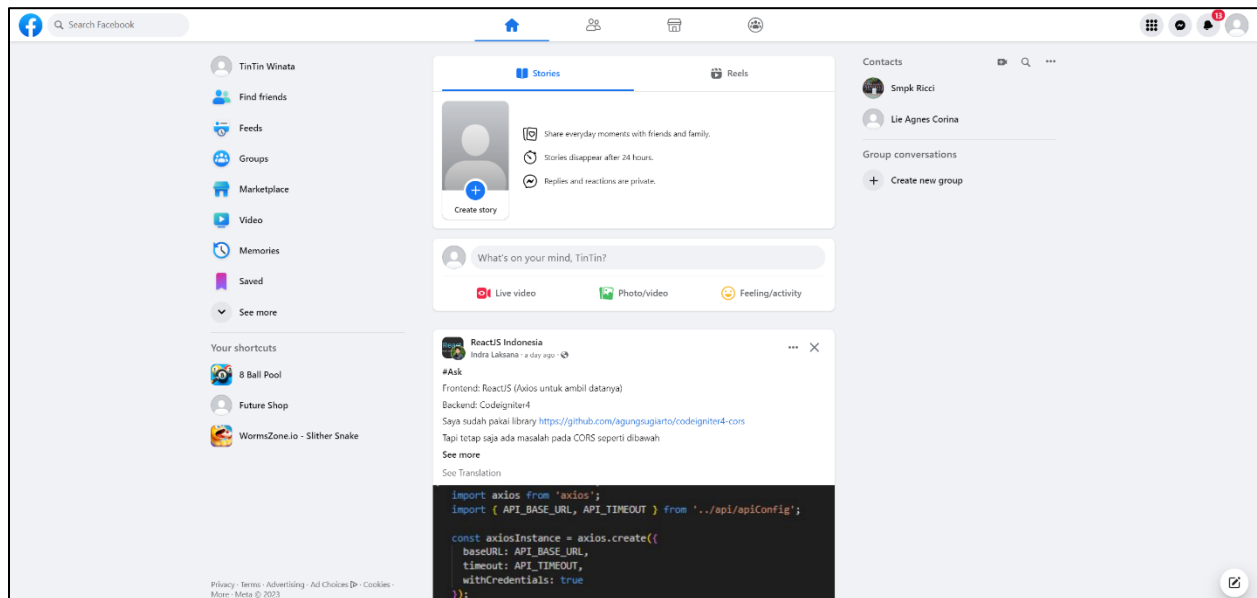
➤ **Home Page**

Figure 5. Home Page

- Must be authorized only to logged-in user.

- Must display the **Post Component** with *infinite scrolling* of user's groups posts, user's friends posts, and public posts.

- Must display **loading animation** or **post component skeleton** when *infinite scrolling* is fetching its nextpost(s).

- Must be able to display **Create Post Component** as a popup.

- Must be able to redirect user to **Create Story Page**.

- Must be able to redirect user to **Reels Page**.

- Must be able to redirect user to **Story Page**.
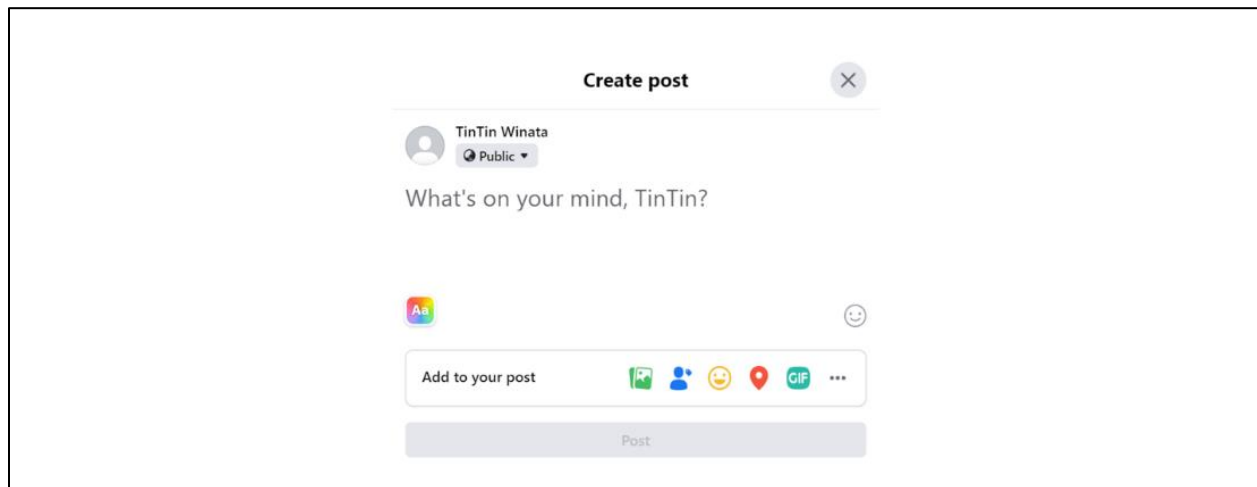
➢ **Create Post Component**

Figure 6. Create Post Component

- Must be able to create posts that contain **text**, **photos**, and **videos**.

- If the **photos** or **videos** are to be added but **photos** or **videos** have been added before, **combine** the **existing ones** with the **new ones** that were previously added.

- Must be able to **remove** added photos or videos.

- Must display text content as **Rich Text**.

- Must be able to display appropriate **autocomplete** when typing **non-normal text**.

- Must be able to display added **photo** and **video**.

- Must be able to **tag** people in the post.

- Must only be able to post if the text content is **not empty**.

- Users can change post privacy to make the post visible to **public**, **friends**, or **specific friends**.
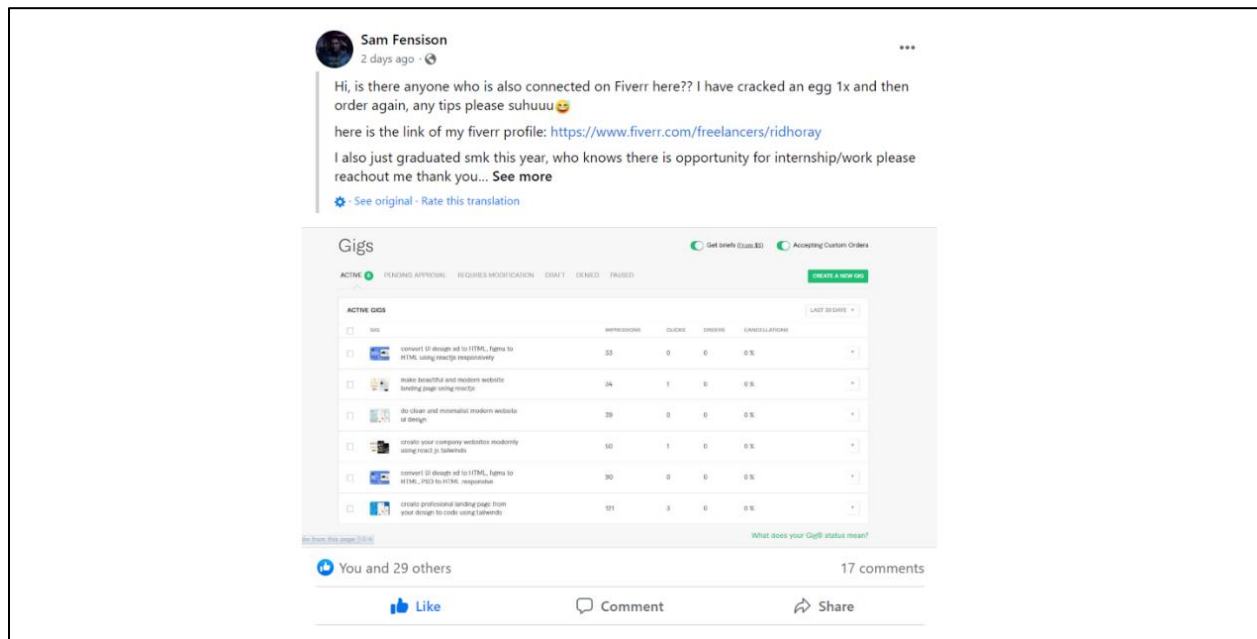
➢ **Post Component**
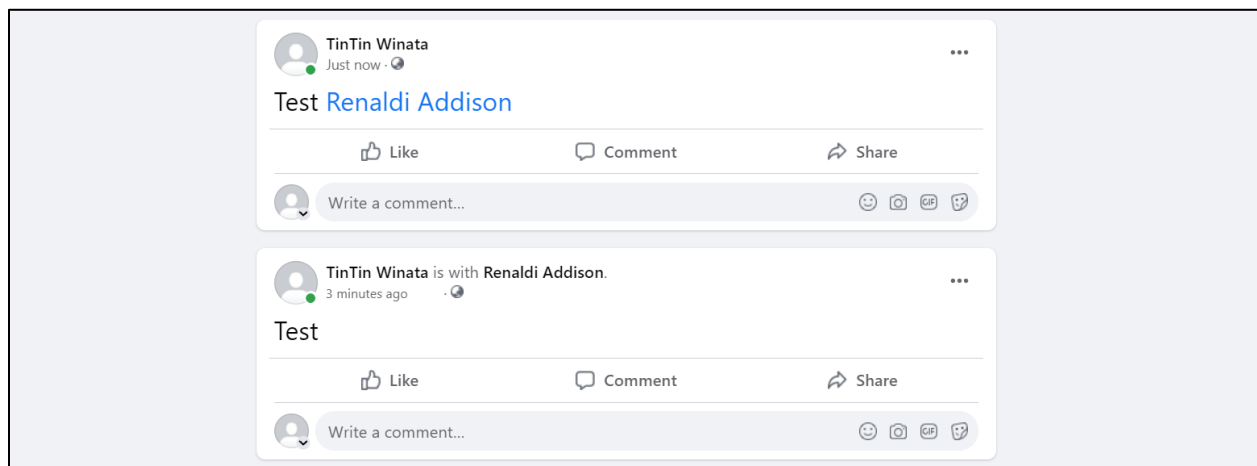

Figure 7. User's Post Component

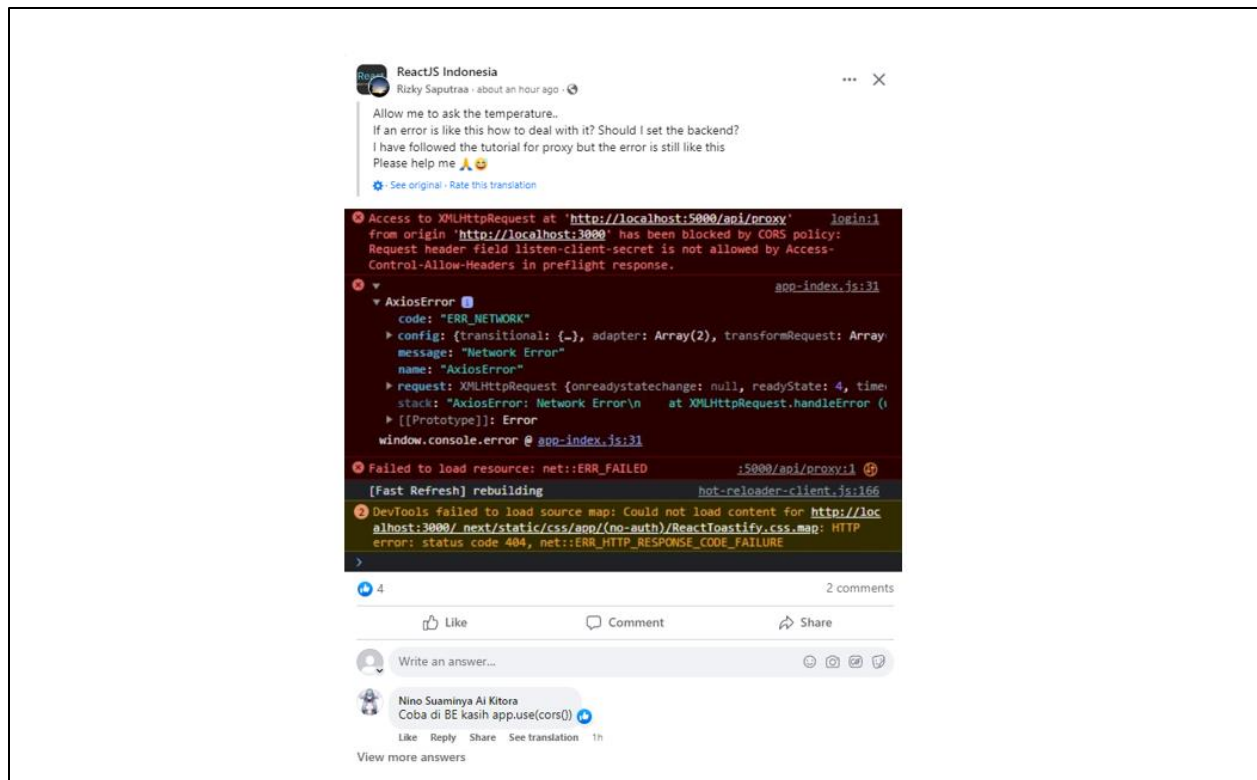*Figure 8. Mention (top) & Tag (bottom)*

Figure 9. Group's Post Component

- Must display text content as **Rich Text**.

- Must be able to display **photos** and **videos** if exist.

- Must display the **profile** of poster (either user or group).

- Must be able to **hover the profile** of poster to display additional information about the user.

- Must be able to be **liked**.

- Must be able to be **commented**.

- Must be able to be shared with other people by **Messenger Page**.

- Must display **number of comments**.

- Must display **number of shares**.

- Must display **number of likes**.

- Must be able display all comments as **Comment Component** with button to load all comments and replies.

- The owner of the post must be able to delete the post.

➢ **Comment Component**

- Must display the **profile** of **commenter** consist of their **name**, and **photo profile**.

- Must be able to be **liked**.

- Must be able to be **replied**.

- Must display **number of likes**.

- Must display **number of replies**.

- Must display all comment replies as **Comment Component** with button to view all comment replies.

➢ **Rich Text**

- URL must have different color from **Normal text** and **clickable**.

- User tag must have different color from **Normal text** and when clicked will **navigate** to the profile of tagged user.

- Hash tag must have different color from Normal text and when **clicked** will navigate to **Search Page** to search all post with such hash tag.

- @ must be used to denote **User** mention (ex: @ren).

- # must be used to denote **Hash** tag (ex: #BINUSKEREN).

- User tag and hash tag must **not contain** any space.

- Must display **additional information** about the **user in user tag** when hovered.

- Normal text is any character(s) that is not part of URL, User tag, or Hash tag.
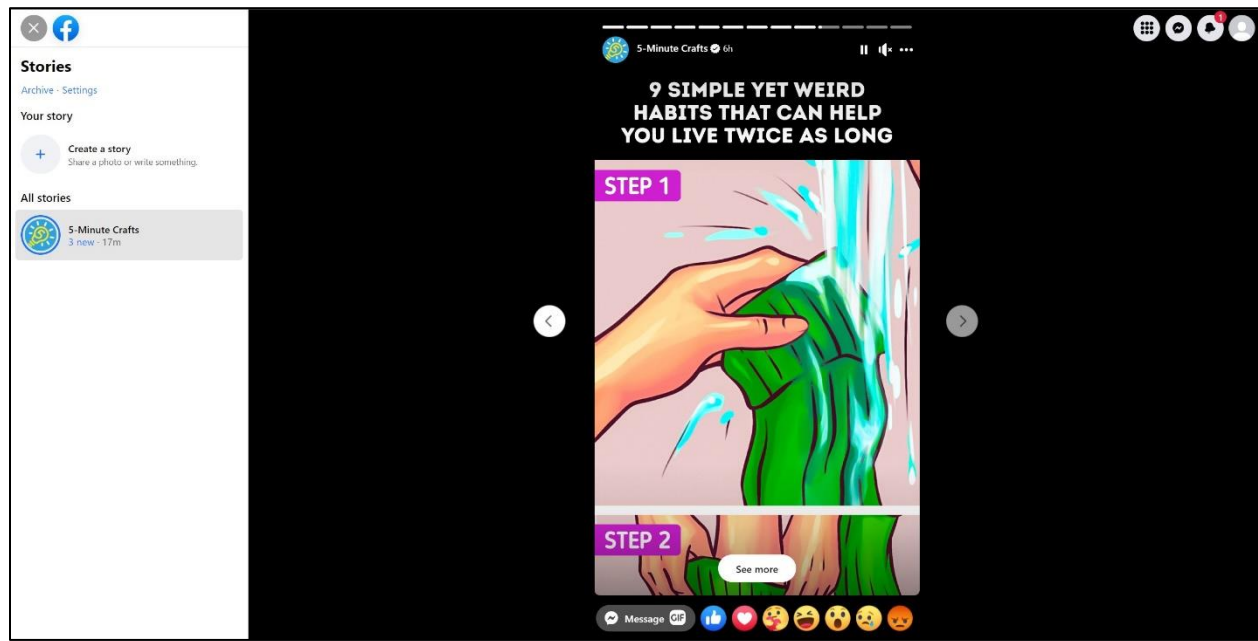
➢ **Story Page**

Figure 10. Story Page

- Must be **authorized** only to logged-in user.

- Each image must be **displayed** within 5 seconds.

- Must be able to **view next** and **previous story**.
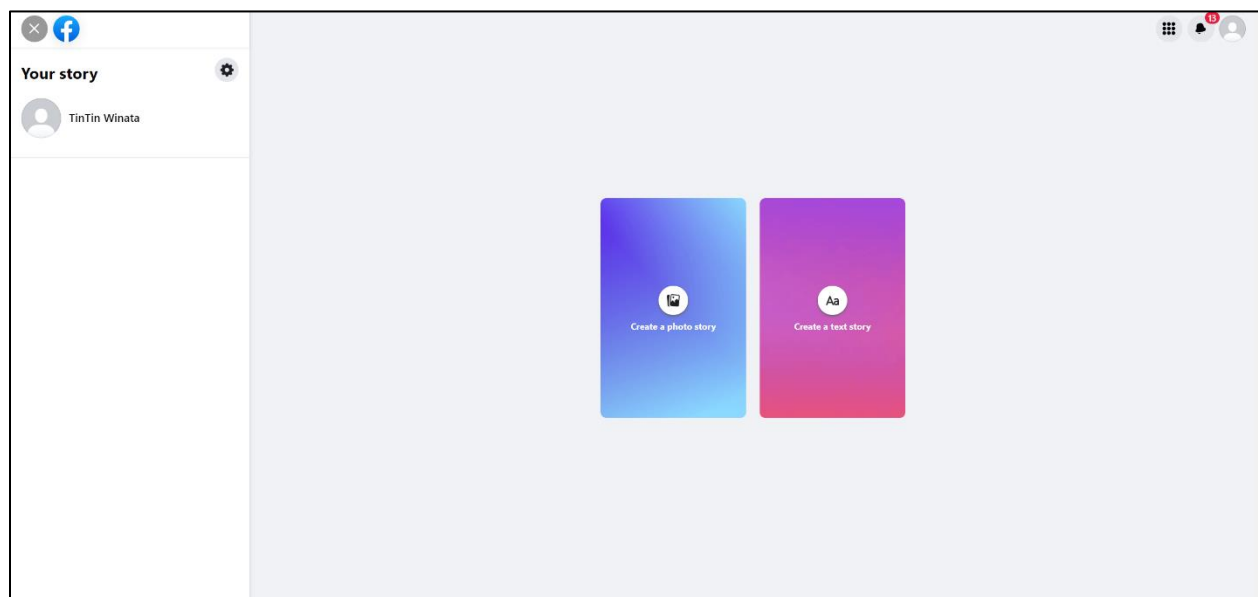
➢ **Create Story Page**

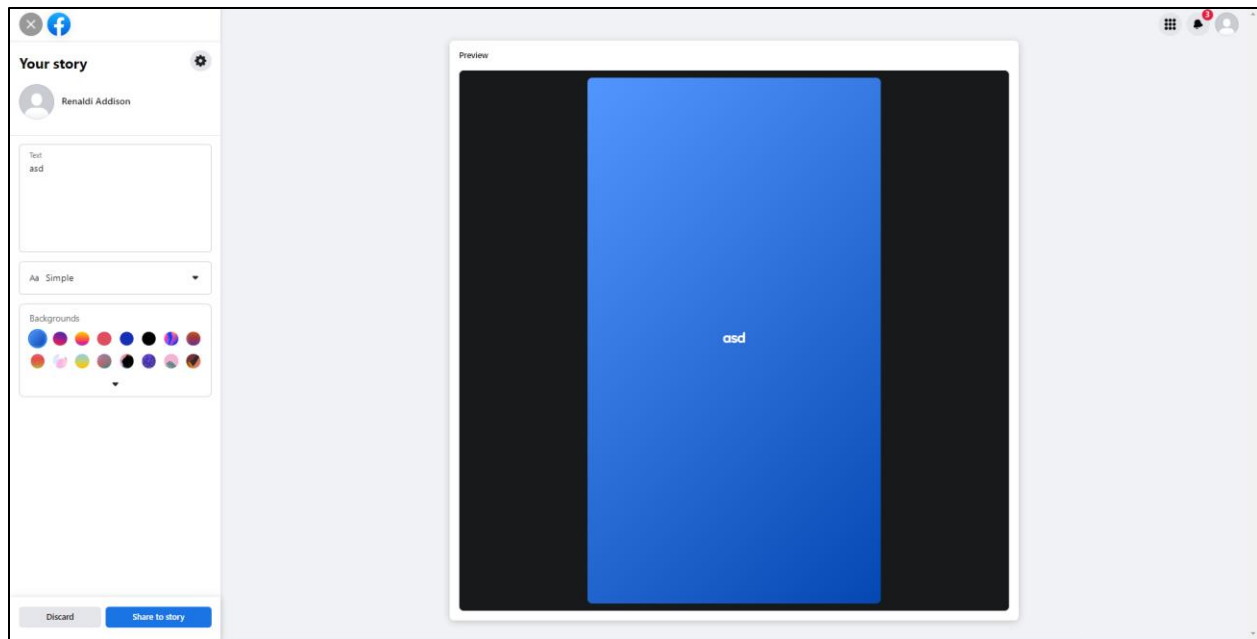

Figure 11. Create Story Page

Figure 12. Create Story Page (Text Story)

- Must be **authorized** only to logged-in user.

- Must be able to create a **photo story** or **text story**.

- Must be able to **edit text's font and background color**.

- The stories that have been created will be **deleted after 24 hours** from their created time.

- The story must be visible to **friends**.

➢ Search Page

Figure 13. Search Page

- Must be **authorized** only to logged-in user.

- Must display **loading animation** or *skeleton* when searching.

- Must be able to display **all posts**, **groups**, **marketplaces**, and **users matching** the search **query** with infinite scrolling.

- Must be able to filter to either **posts**, **groups**, **marketplaces**, or **user**.
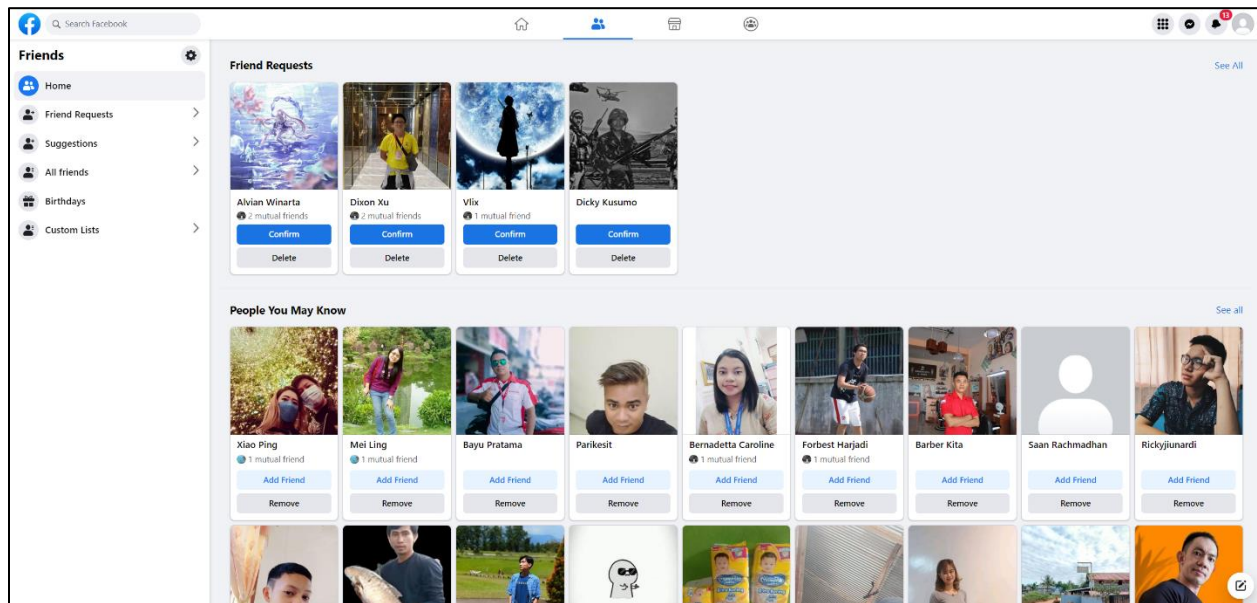
➢ **Friends Page**

Figure 14. Friends Page

- Must be **authorized** only to logged-in user.

- Must display **all friend** requests to currently logged in user.

- Must be able to **confirm** or **delete friend requests**.

- Must display **People You Might Know Section**.

➢ **People You Might Know Section**

- Must display all possible **user suggestions**.

- Suggestions can be obtained by placing **A** as the current logged in **user**, **B** as the user that is connected to **A**, and **C** as the user that is connected to **B** but not **A**. Then, **display C**.

- Must be able to **add friends** or **remove suggestions**.
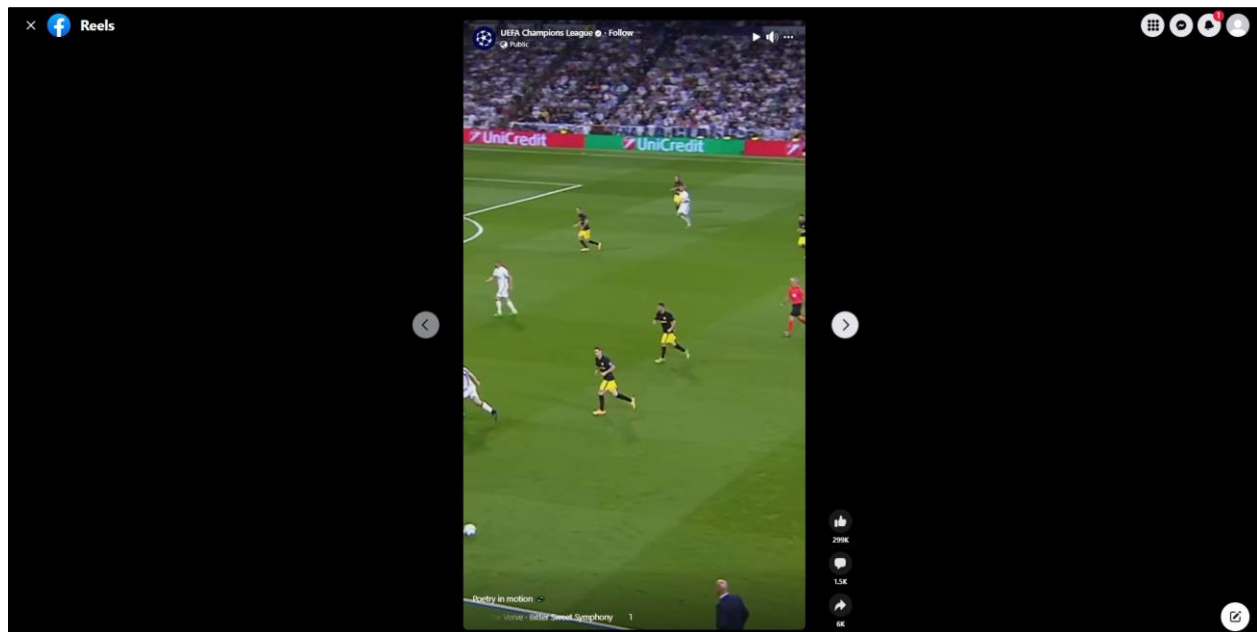
➢ **Reels Page**

Figure 15. Reels Page

- Must be **authorized** only to logged-in user.

- Must display text content as **Rich Text**.

- Must display **all** public reels.

- Reel's video must be displayed in **portrait format**.

- Must be able to be **liked**.

- Must be able to be **commented**.

- Must be able to be **shared** with other people by **Messenger Page**.

- Must display **number of comments**.

- Must display **number of shares**.

- Must display **number of likes**.

- Must be able display all comments as **Comment Component** with button to load all comments and replies.

- Must be able to redirect user to **Create Reels Page**.
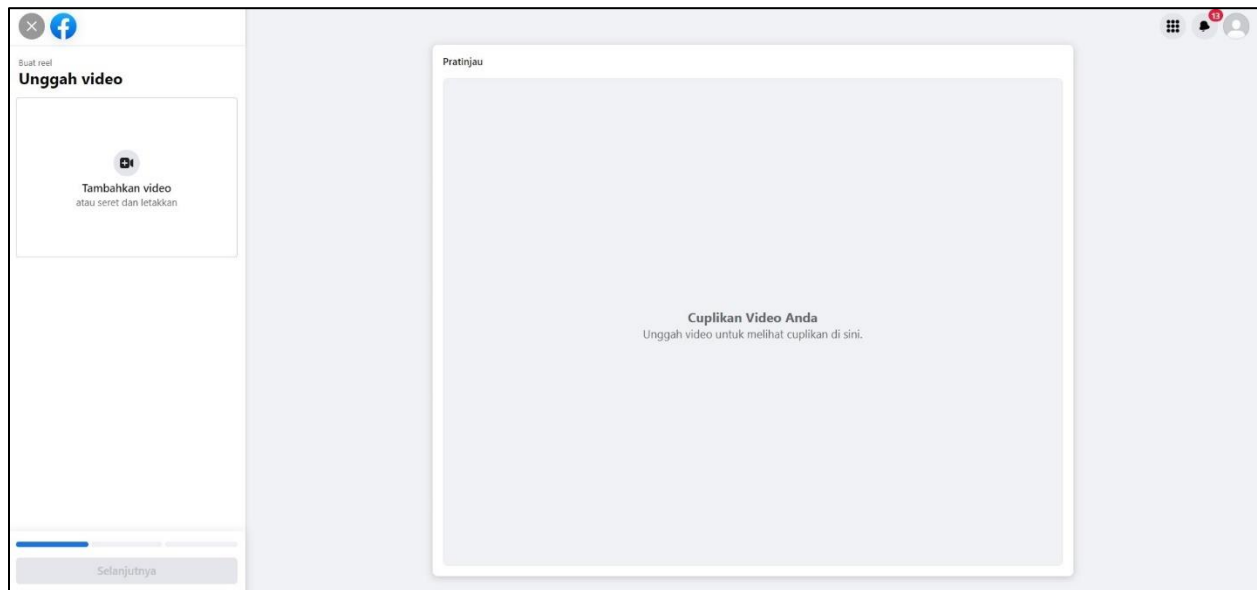
➢ **Create Reels Page**

Figure 16. Create Reels Page

- Must be **authorized** only to logged-in user.

- Reel's video must be **between 1 – 60 seconds**.

- If a new video is to be added but there is an existing video that has been added before, **remove** the existing one and **replace** it with the new one.

- Must be able to **display** the added video.

- Must be able to **remove** added video.

- Must display text content as **Rich Text**.

- Must be able to **display appropriate autocomplete** when **typing non-normal text**.

- Must only be able to post the reels if the **text content** is not empty.

- Reels must be visible to **public**.
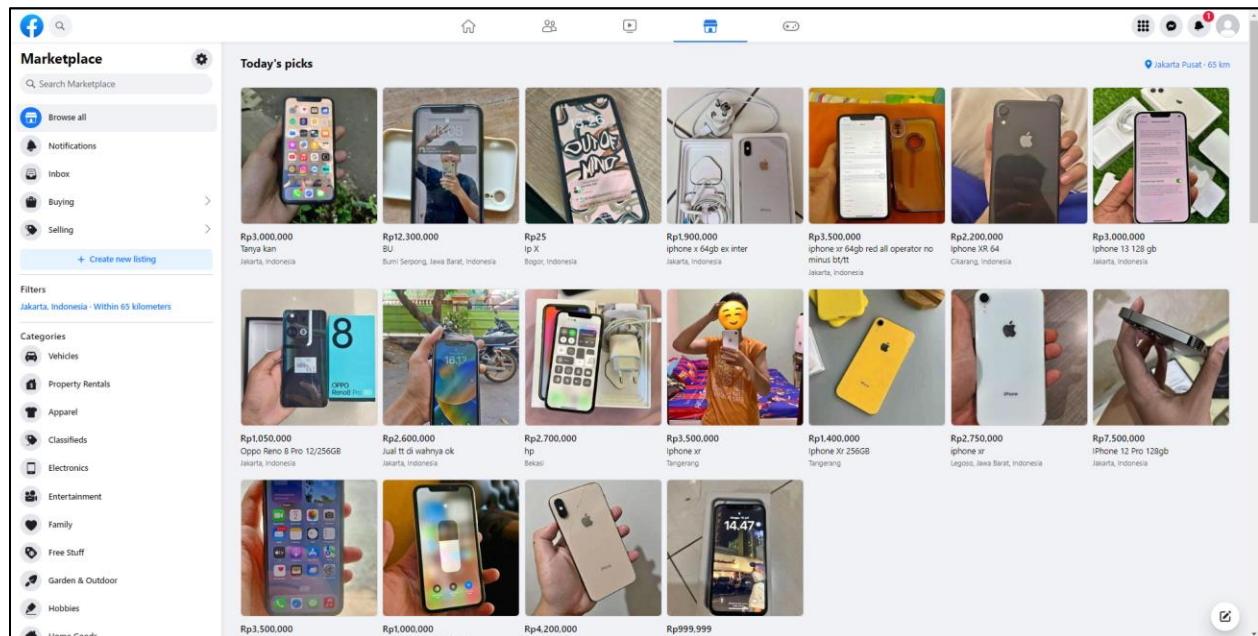
Marketplace Page

Figure 17. Marketplace Page

- Must be **authorized** only to logged-in user.
- Must display the **Marketplace Item Component** with *infinite scrolling* of **items**.
- Must display loading animation or marketplace item **component skeleton** when *infinite scrolling* is fetching its nextitem(s).
- Must be able to redirect user to **Create Marketplace Item Page**.

➢ **Marketplace Item Component**

Figure 18. Marketplace Item Component

- Must display one item's **image**, **price**, **title**, and **seller's location**.
- Must be able to redirect user to **Marketplace Item Page**.

> **Marketplace Item Page**

Figure 19. Marketplace Item Page

- Must be **authorized** only to logged-in user.
- Must display all item's **image**, **title**, **price**, **description**, **condition**, **seller's location**, **seller's name,** and **seller's image**.
- Must be able to be shared with other people by **Messenger Page**.
- Must be able to **send messages** with **item's seller**.
- Must be able to **hover** the **profile** of the seller to **display additional information** about **the seller**.
- Must be able to display **Seller Profile Component** as a popup.
- The owner of the marketplace item must be able to **delete** the item.
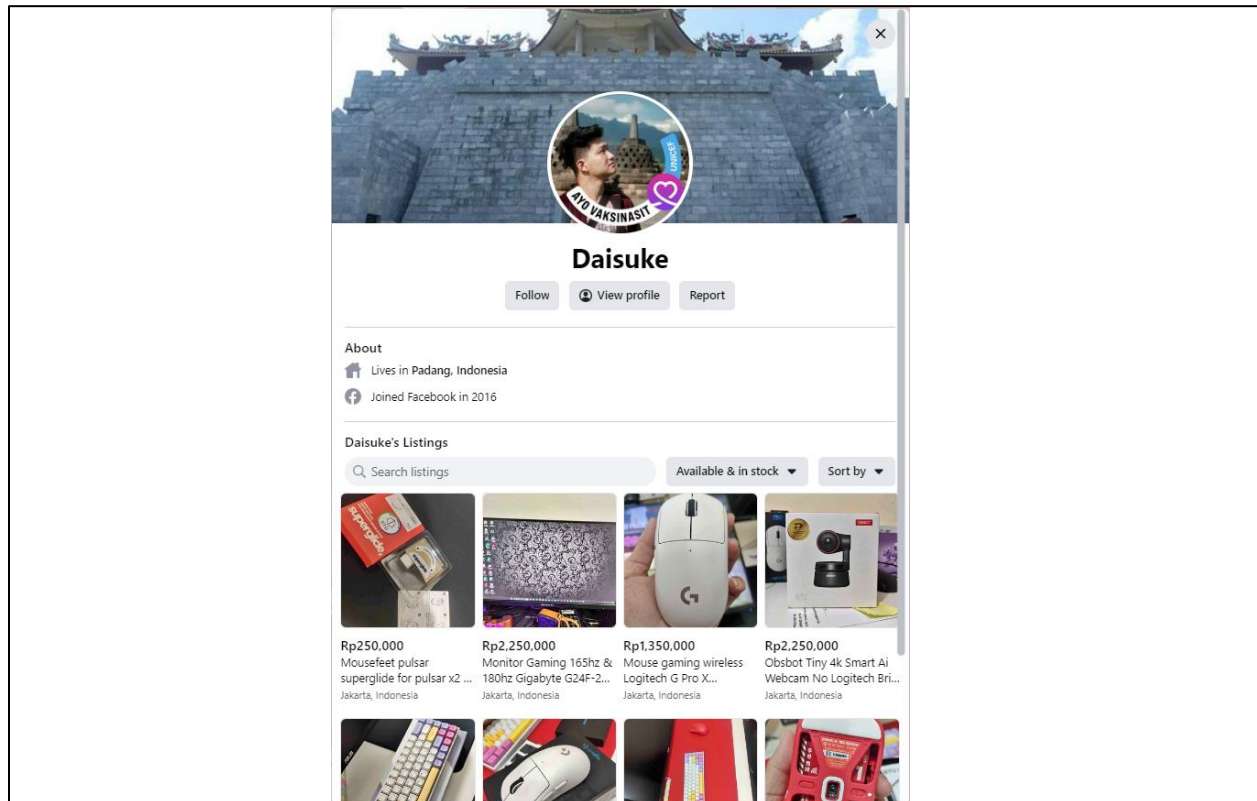
➢ **Seller Profile Component**

Figure 20. Seller Profile Component

- Must display seller's profile **picture**, **banner**, and **name**.

- Must display the **Marketplace Item Component** of all seller's items.

- Must be able to **search** seller's item.

- Must be able to **sort seller's** item by **newest**, **oldest**, or **title** (A-Z).

➢ **Create Marketplace Item Page**

Figure 21. Create Marketplace Item Page

- Must be **authorized** only to logged-in user.

- Must be able to **input photos** (**up to 10**).

- If the **photos** are to be added but there are photos that have been added before, **combine** the existing ones with the new ones.

- Must be able to **display photo or video** when added.

- Must be able to **remove** added photos.

- Must register a new item using **title**, **price**, **description**, **condition**, and **seller's location**.

- Must only be able to post items if the text content is **not empty**.
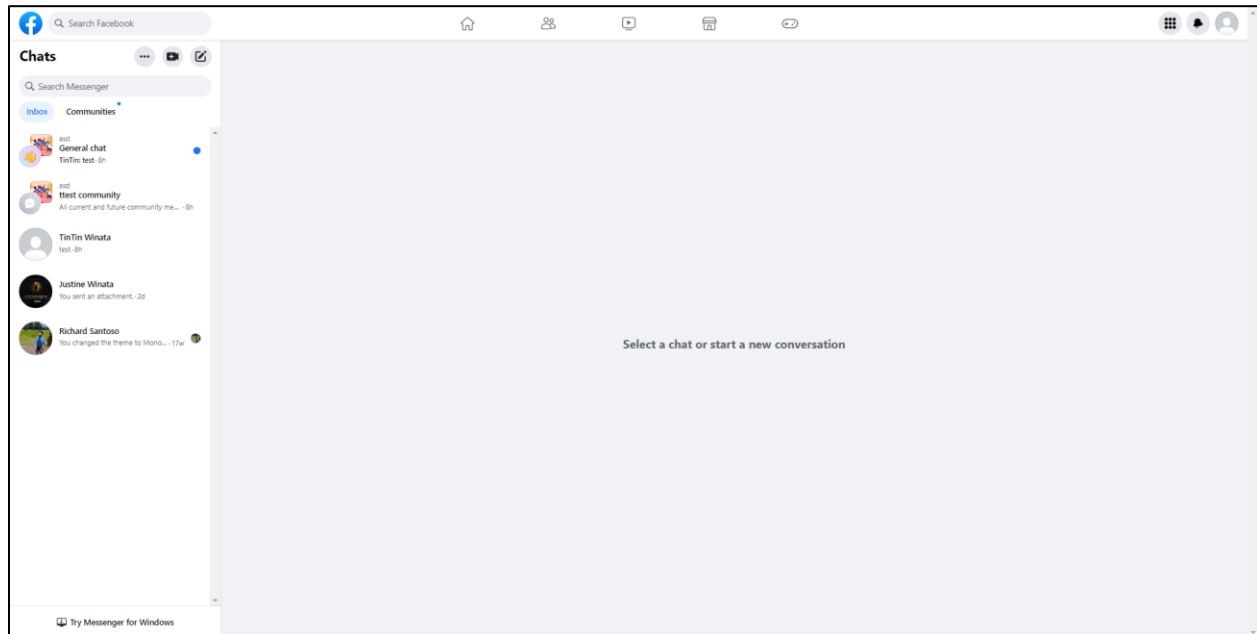
➢ **Messenger Page**

Figure 22. Messenger Page

- Must be **authorized** only to logged-in user.

- Must **display all user's personals** or **groups** message history.

- Must be able to **search chats and users**.

- Must be able to **send either text message** or **voice clip**.

- Must be able to display messages in **real time**.
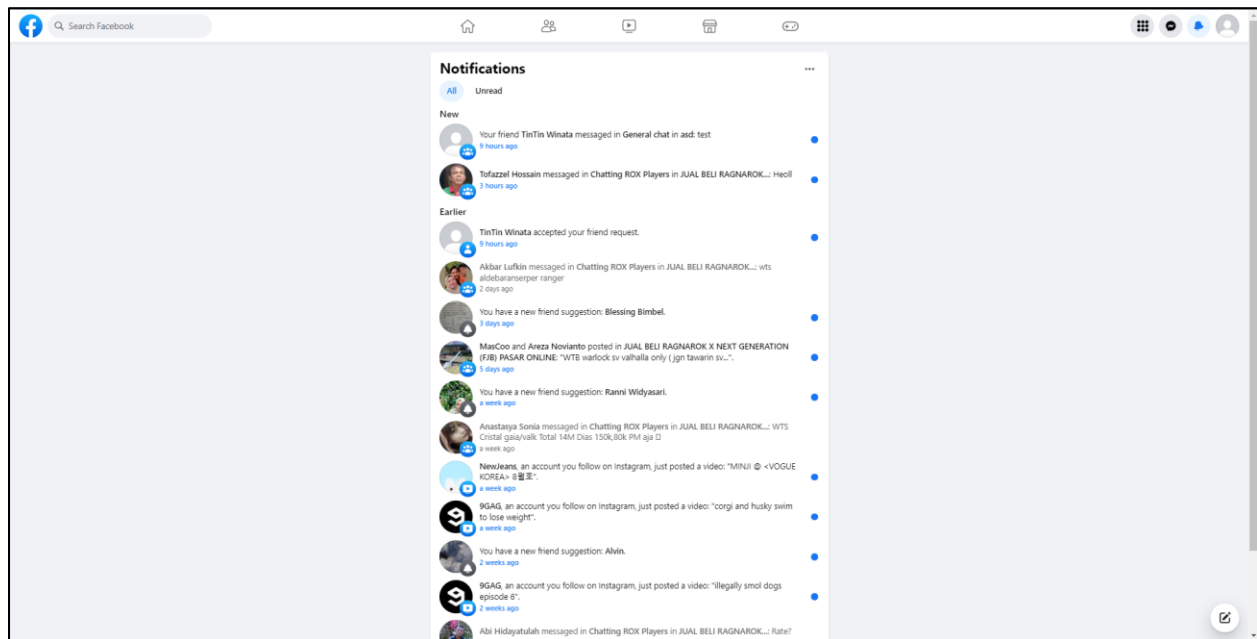
➢ **Notification Page**

Figure 23. Notification Page

- Must be **authorized** only to logged-in user.
- Must be able to **display all at least 7 activity** that has been done by currently logged-in user and all user that has been connected to currently logged in user. Please **don't forget** to choose the activity **wisely**, simple activities such as logging in will **not be considered** and included in the assessment.
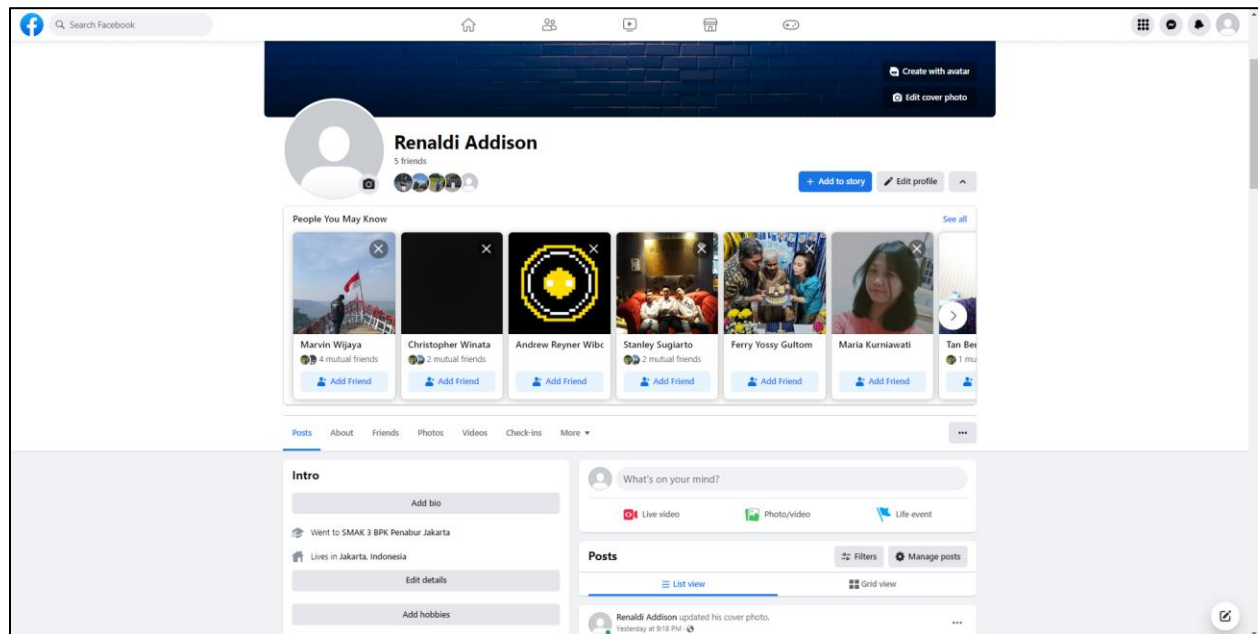
➢ **User Profile Page**

Figure 24. User Profile Page

- Must be **authorized** only to logged-in user.

- Must display **People You Might Know Section**.

- Must display **at least 5** current user's personal **information**.

- Must display **number of friends**.

- Must display **number of mutuals**.

- Must be able to send **add friend request** if the profile is not the current logged in user.

- Must be able to **edit own's personal information** if the profile is the current logged in user.

- Must be able to **send messages** if the profile is **not** the current logged in user.

- Must be able to be **shared** with other people by **Messenger Page**.

- Must be able to **block user** (block all notification from the blocked user).

- Must display a section to display all **user's posts**, **friends**, and **reels**.
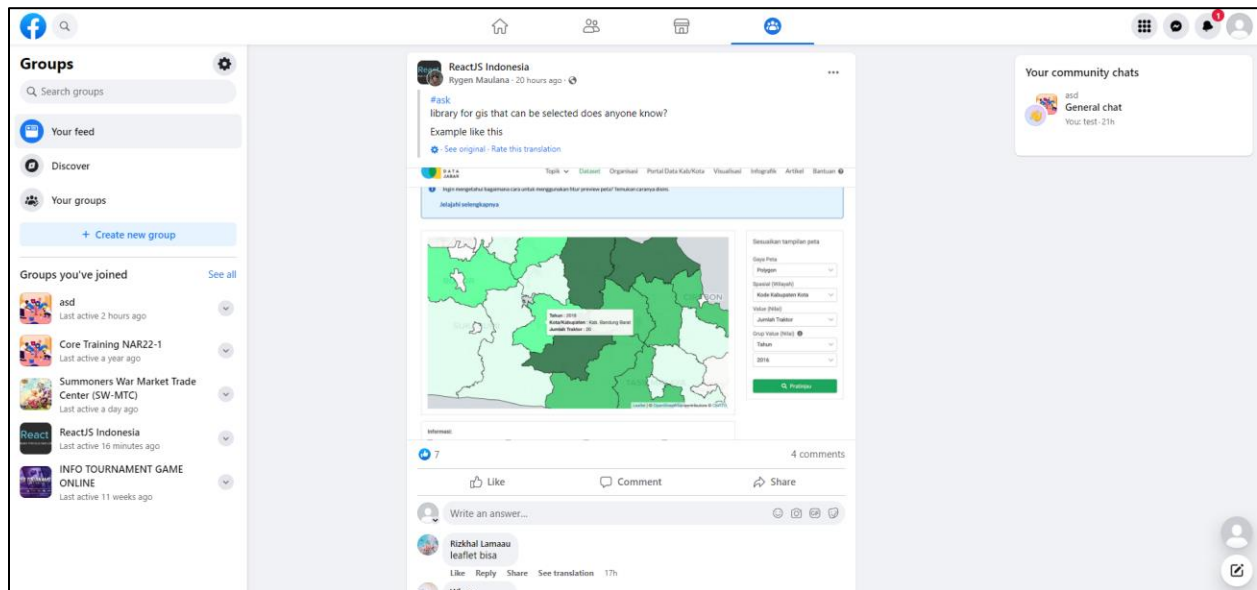
➢ **Group Page**

Figure 25. Group Page

- Must be **authorized** only to logged-in user.

- Must be able to **search public groups**.

- Must display all **joined groups**.

- Must display the **group Post Component** with infinite scrolling of user's joined groups' posts.

- Must be able to redirect user to **Create Group Page**.
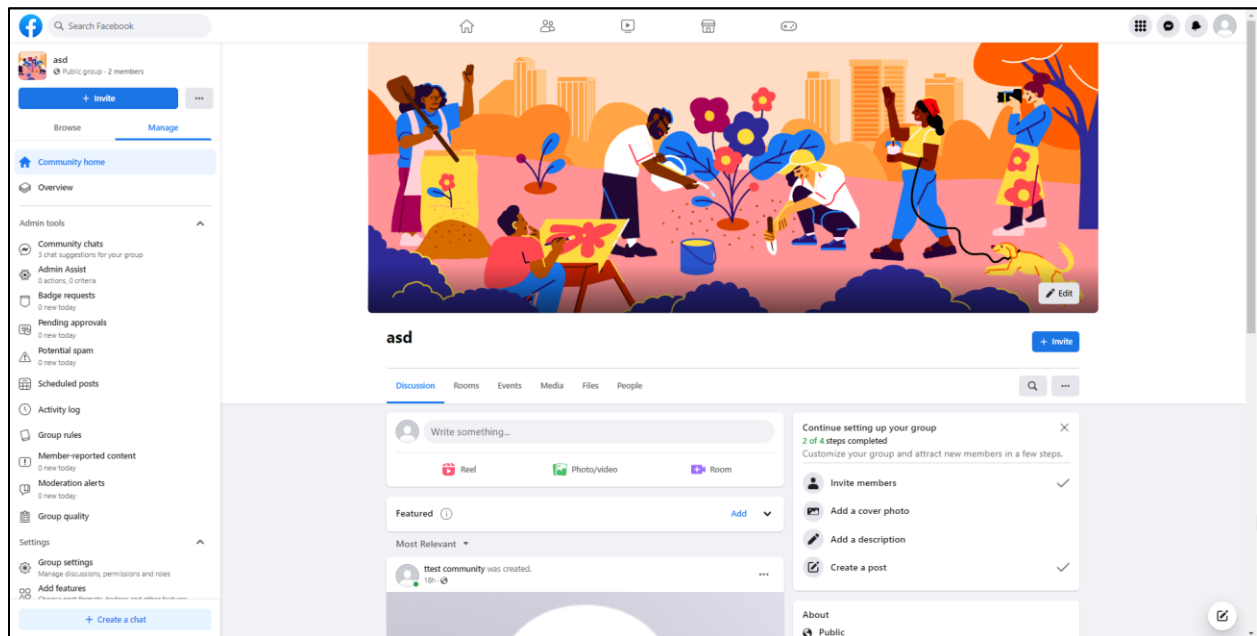
➢ **Group Profile Page**
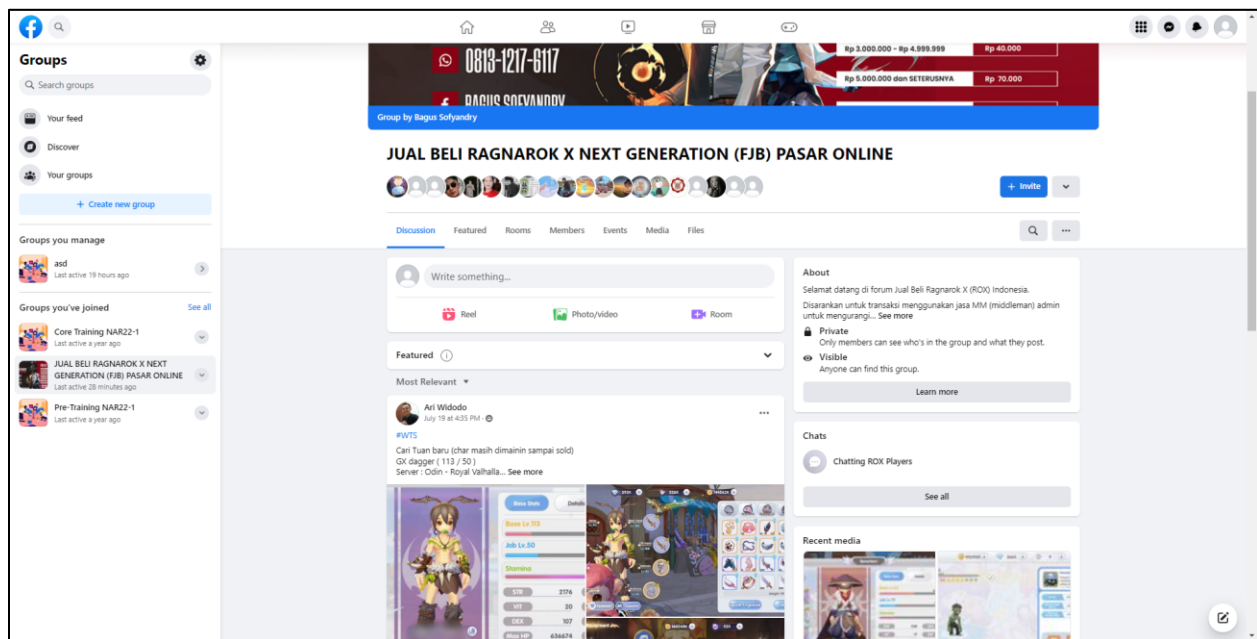

Figure 26. Group Profile Page (admin role)

Figure 27. Group Profile Page (member role)

- Must be **authorized** only to logged-in user.

- Must display at **least 3 group informations**.

- Group must have **2 roles** (admin and member).

- All roles must be able to **create post** and display **Create Post Component** as a popup.

- All roles must be able to **invite their friends** to the group.

- Must be able to **send join group request** if currently logged-in user isn't an **admin** or **member** in the group.

- Must be able to **leave group** if currently logged-in user is an admin or member in the group.

- If there is only **one admin** and the group has another member, the admin cannot **leave the group** unless there is another admin.

- Must display a section to display **all group's posts** and **chat rooms**.

- Must be able to be shared with other people by **Messenger Page**.

- **The group** will be deleted after all the **participants** leave the group.

- Admin role:
  - Must be able to **create chat rooms** (a group can have multiple chat rooms).
  - Admin will always be a **member of a chat room**.
  - Must be able to **invite group members to a chat room**.
  - Must be able to **promote** another group members to admin role.
  - Must be able to **kick** another group members.
  - Must be able to **approve** join group request.
  - Must be able to **edit** the group's banner picture.
  - Must be able to **delete** another member's posts.
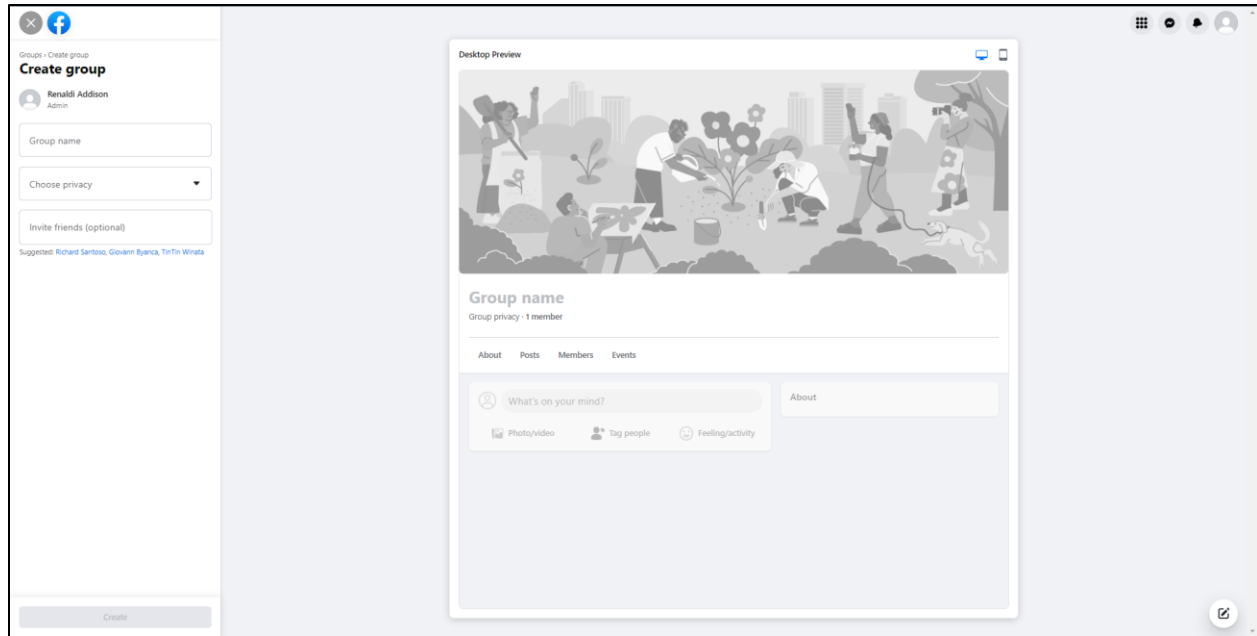
➢ **Create Group Page**

Figure 28. Create Group Page

- Must be **authorized** only to logged-in user.

- Must only be able to **create a group** if the text content is not empty.

- Users can change the group's privacy to make the group visible to **public** or **private**.

Note:

- All features that have been made that are not listed will not be graded.