General requirements:

- Displays on current versions of Firefox
- Displays on current versions of Google Chrome
- Main navigation links are clearly and consistently labeled
- The color scheme is limited to a maximum of three or four colors plus neutrals
- No JavaScript/react errors are generated
- All internal hyperlinks work

Features requirements

- A user is considered online if:
  - They have the app opened in the browser, and
  - They are logged in

Features		
#	**Feature**	**Minimum Requirements**
1	User registration(0)	The registration process should capture at least: ● Username ● Password ● At the end of the registration process, the app should display a message to confirm the registration or why the registration was unsuccessful. ● Newly registered accounts should be redirected to the login view ● The new account should be visible in the database ● The registration date should also be recorded in the database ● You should validate the password: length/range / format and type ● The password should be encrypted in the database

2	Login/Auth (0)	The login view should require at least: ● Username ● Password ● If the authentication is successful, the user should be sent to **the main view** ● If the authentication is unsuccessful, the user should remain on the login page, and the reason for the failure should be displayed ● You should implement an account lockout rule for too many unsuccessful attempts (> 3). ● The lockout should expire after some time (for example, 30 minutes) ● The user should be able to reset their password if forgotten
3	User profile page(0)	The user profile should display at least: ● The username ● The registration date ● An avatar or user profile The view should provide the ability to: ● Change password ● Deactivate the account ● [optional] change avatar / profile picture You are not required to delete all information related to a deactivated account from the database. However, the account (and posts/messages) should not be visible in the app
4	Create a public group (0)	From the main view, the user should be able to create new group. The group creation form should ask for (at least): ● The type of the group (public or private) ● A list of topics discussed in the group

5	Create a private group (0)	From the main view, the user should be able to create new group. The group creation form should ask for (at least): ● The type of the group (public or private) ● A list of topics discussed in the group
6	Add / remove administrators (0)	The **group page (board)** should have an option for group administrators to: ● Promote a group member as administrator ● Revoke the administrator status ● The creator of the group is automatically the group administrator ● The creator of the group cannot be removed as administrator ● Members should be notified when they are promoted to admin status ● The group page should list all the admins
7	Request and join a public group (0)	**The main view** should list all the public group and should have an option for users to request admission into a group ● While the request is pending the user cannot send another request to the same group ● If the request is denied the user must be notified. The user send another request to the same group ● If the request is granted, the group should be added to the list of the groups that the user belongs to ● Only admins can approve membership requests

8	Invite a user into a group (public and private) (0)	From the **main view or the group board**, members should be able to: ● Invite other users to join a group (they must know the username of the user) ● The user should get a notification that they are invited to join a group ● The invited user can approve or decline the invitation ● If the invitation is approved, then the admins of the groups are notified so they can grant access. The status of the group should be pending while the request is being processed ● If the invitation is declined, then the members inviting the user must be notified
9	Leave a group (0)	From the group board, members should be able to: ● Leave a group ● Admins should be notified that a member left the group ● The group should not be in the list of the groups that the user belongs to ● All the posts of the users should still be listed on the group board ● Members should not be able to interact (message) with the user who left the group (unless they are both in another group) ● The old messages should not be deleted but sending a new message should not be possible
10	Filter groups by topics (tags) (0)	From the main view, users should be able to filter public groups by topics.

11	Post into a group (0)	The group board should allow users to create and post microblogs (posts). ● Implement a maximum size for a microblog ● Implement a maximum size policy for the media attachments ● Microblogs can have attachments (images, audio, video)
12	Flag a post for deletion (0)	From the group board, members should be able to flag a post for deletion. ● Group admins should be notified ● The admins should know who flagged the post ● The author of the post should not know who flagged the post ● The author of the post should only be notified if the post is deleted (by the admins) ● If the post is not deleted the user who flagged the post should be notified ● If the post is deleted the user who flagged the post should be notified
13	Delete a post (0) author and admins	The user should be able to: ● Select and delete a specific post that they authored ● Admins can delete any posts (flagged or not) ● A deleted post (and comments) should disappear from the board ● Deleted posts cannot be restored ● Deleted posts should be removed from the database or flagged as removed (in the DB) if not deleted ● The author of the post should be notified if the post was deleted by the admins

14	Show list of public groups (0)	The main view should: ● List all the public groups ● There should be a distinction between the groups that the user is a member of and the other ones ● Private groups (that the user belong to) should ba labels as such
15	Display posts in a group (0)	● The app should display posts of contacts from the most recent to the least recent ● Media attachments should be played within the app
16	Comment (reply) on a post (0)	A user should be able to comment on a post ● The comment should be displayed below the original post ● Comments should be displayed in descending order of creation (newer post first)

17	Interactive API documentation using Swagger (0)	Swagger is an online tool used to design and document APIs (RESTful). ● You will use swagger to build an interactive documentation of your API (https://swagger.io/)
18	Hide a member's post (1)	A user should be able to hide a post on the group board ● The hidden post should disappear from the users page ● Refreshing or reloading the page should not make the post visible ● This action cannot be undone
19	Edit/Delete a comment (1)	A user should be able to to delete a comment ● The user can only delete a comment they authored ● The comment should disappear from the app ● Deleted comments cannot be restored ● Deleted comments should be removed from the database or flagged as removed if not deleted

20.0	Send/receive private text message (1) - asynchronously	The user should be able to select a contact, and a new view (**messaging view**)should be displayed: ● The **messaging view** should allow users to type and send messages ● The receiver does not need to be online ● The receiver of the message will receive the message when they are logged in ● When the receiver logs in, the message should be visible in their message view ● The message view of the sender should display the most recent messages with the selected contact ● A user cannot send a message to themself ● A user can only message people (one or many) with whom they share at least one group ● A user cannot find or send a message to a deleted account ● A user can send a message to a locked-out account
20.1	Send/receive private audio message (1) - asynchronously	The user should be able to select a contact, and the **messaging view** should be displayed: ● The user should be able to **send prerecorded audio files** located on their device ● The receiver does not need to be online ● The receiver of the message will receive the audio when they are logged in next ● When the receiver logs in, the message should be visible in their message view ● The message view of the sender should display the most recent messages with the selected contact ● The audio message should be played within the app ● A user cannot send an audio message to themself ● A user can only message people (one or many) with whom they share at least one group ● A user cannot find or send an audio message to a deleted account ● A user can send an audio message to a locked-out account ● Implement a maximum size policy for the audio files

20.2	Send/receive image message (1) - asynchronously	The user should be able to select a contact, and the **message view** should be displayed: ● The message view should allow users to **send an image file located on their device** ● The receiver does not need to be online ● The receiver of the message will receive the image when they are logged in next ● When the receiver logs in, the message should be visible in their message view ● The message view of the sender should display the most recent messages with the selected contact ● The image should be displayed within the app ● A user cannot send an image to themself ● A user can only message people (one or many) with whom they share at least one group ● A user cannot find or send an image to a deleted account ● A user can send an image to a locked-out account ● Implement a maximum size policy for the image file
20.3	Send/receive private video message (1) - asynchronously	The user should be able to select a contact, and the **message view** should be displayed: ● The message view should allow users to **send a video file located on their device** ● The receiver does not need to be online ● The receiver of the message will receive the video when they are logged in next ● When the receiver logs in, the message should be visible in their message view ● The message view of the sender should display the most recent messages with the selected contact ● The video should be playable within the app ● A user cannot send a video to themself ● A user can only message people (one or many) with whom they share at least one group ● A user cannot find or send a video to a deleted account ● A user can send a video to a locked-out account ● Implement a maximum size policy for the video file
21	Provide several ways of sorting groups in the main view: newest	From the main view (in addition to filtering) the users should be able to sort public groups by: ● newest message

	message, number of posts, number of members (2)	● number of posts in the group, ● number of members
22	Support for Hashtags filtering in posts and comments (2)	From the group board, users should be able to filter posts and comments using “hashtag handles.”
23	@mentions in posts and comments (2)	When typing a message, the user should have the ability to ● Tag one or more members in a message (@contact_name syntax) ● Only group members can be mentioned
24	Groups suggestions (3)	● The app should suggest new public groups (to join) ● The suggestion can be implemented in the main view or another one ● There is no requirement on how many new groups to suggest or how to select the groups (a random pick is acceptable)
25	Live update (3) - posts / messages	● If the user is on the app carrying on a conversation, then new messages should be displayed automatically and within 5 seconds (after being sent) ● If the user is on the app, then new posts should be automatically displayed and within 5 seconds (after being posted) ● The user does not need to refresh the page for the app to update ● You will use either long-polling or server push to implement this feature
26	Notifications(3) - posts / messages	● The user should be notified (within the app) whenever any of the following events happen: new message, new mention in a message or post, a new post, message deleted ● The notification should be automatic whenever the app is launched (in the browser) ● The user does not need to refresh the page for the notification to be displayed. ● You will use either long-polling or server push to implement this feature.

27	Delivery / Read receipts (3)	In the messaging view, the app should notify the user when a message (that they sent) has been: ● Delivered: the message is delivered to the recipient's browser, but it hasn't seen it. ● Read: the recipient has read your message or seen your picture, audio file, or video.
28	Analytics for groups (4)	Analytics should include (but not limited to): ● Number of members ● Number of posts ● Number of posts deleted ● Number of post flagged ● Number of post hidden
29	Pagination (4)	When displaying posts, the app will use pagination ● https://uxplanet.org/ux-infinite-scrolling-vs-pagination-1030d29376f1
30	Infinite scroll (4)	When displaying posts, the app will use infinite scroll ● https://uxplanet.org/ux-infinite-scrolling-vs-pagination-1030d29376f1
30	Make your app responsive - adapt to smaller (mobile) screen sizes (4)	Your app should adapt to smaller screen sizes ● You can use CSS to make your app responsive