General requirements:

- Displays on current versions of Firefox
- Displays on current versions of Google Chrome
- Main navigation links are clearly and consistently labeled
- Color scheme is limited to a maximum of three or four colors plus neutrals
- No JavaScript 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 the user 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 of the failure should be displayed ● You should implement an account lockout rule for too many unsuccessful attempts (> 3). ● The lockout should expire after a time period (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 messages) should not be visible in the app
4	Follow/unfollow user (0)	The user should be able to: ● Search and add (follow) new contacts ● The new contact should be visible in the list of contacts ● Post of the new contact must be displayed ● The user should be able to search and unfollow a contact ● The removed contact (and their posts) should not be displayed ● Re-following a previously removed contact should make their posts visible again

5	Block a follower (0)	The user should be able to: ● Search and block a contact ● A blocked contact cannot communicate with their blocker ● A blocked contact cannot see the posts of their blocker ● A blocked contact cannot follow their blocker The user should be able to unblock a contact
6	Post a new microblog / post (0)	The app should have a **view** allowing users to create and post microblogs. ● Implement a maximum size for a microblog ● Implement a maximum size policy for the media attachments
7	Delete a microblog (0)	The user should be able to: ● Select and delete a specific post ● A deleted post should disappear from all followers views immediately ● Deleted posts cannot be restored ● Deleted posts should be removed from the database or flagged as removed if not deleted
8	Display posts (microblogs) of “following” contacts (0)	● The app should display posts of contacts from the most recent to the least recent ● Media attachments should be played within the app
9	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)
10	Hide a contact’s post (0)	A user should be able to hide a contact’s post ● 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

11	Edit/Delete a comment (0)	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
12	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/)
13.0	Send/receive 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 image 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 cannot find or send a message to a deleted account ● A user can send a message to a locked-out account

13.1	Send/receive 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 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
13.2	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 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

13.3	Send/receive 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 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
14	Start a new live stream (1)	A user should be able to broadcast a live stream: ● The stream will use the computer's camera ● Users (followers or not) should be able to watch a live stream ● Blocked users cannot see their blocker's live stream
15	Display how many people are watching a live stream (1)	● The live stream page should display the number of people watching
16	Comment on a live stream (1)	● Users should be able to comment on live stream ● Comment are listed from the most recent to the least recent
17	Show list of live streams (1)	● The app should display a list of live streams ● Users should be able to click on a link to join a live stream page
18	Join (merge) a contact's live stream (2)	Users should have the ability to join a contact's live stream page: ● The user will send a request to join a live stream ● If the request is accepted, then both live streams are displayed on the same page

19	Comment on a merged live stream (2)	• Users should be able to comment on a page displaying more than one live streams
20	Support for Hashtags filtering in posts and comments (2)	User should be able to filter posts and comments using "hashtag handles"
21	@mentions in posts and comments (2)	When typing a message, the user should have the ability to • Tag one or more contacts in a message (@contact_name syntax) • Only the user's contacts can be mentioned
22	Contact suggestions (2)	• The app should suggest new contacts (to add) • The suggestion can be implemented in the main view or another one • There is no requirement on how many new contacts to suggest or how to select the contacts (a random pick is acceptable)
23	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
24	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, new post from a contact, new follower • 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

25	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 the recipient hasn't seen it. ● Read: the recipient has read your message or seen your picture, audio file, or video
26	Analytics for posts (4)	How many people commented on a post (with usernames) How many people hid the post (with user names)
27	Analytics for live streams(4)	How many people viewed a live stream How many people commented on a live stream (with usernames) How long each viewer watched the live stream
28	Pagination	When displaying posts, the app will use pagination ● https://uxplanet.org/ux-infinite-scrolling-vs-pagination-1030d29376f1
29	Infinite scroll (4)	When displaying posts, the app will use infinite scroll ● https://uxplanet.org/ux-infinite-scrolling-vs-pagination-1030d29376f1