

- Project Features List
 - Newsfeed presentation
 - Offers image preview of articles, and large headline, in order to maximize outrage, and minimize
 - Ideally offers infinite scroll, but might be too hard
 - Newsfeed articles
 - Assortment of questionable media outlets and articles. Each post has buttons to like, comment, and share.
 - User Profile
 - Each profile consists of a user picture and bio section. Here the user can also add keywords to their “Interests” to help with sorting articles within their unique newsfeed.
 - Search Bar
 - Searches database for articles relating to search criteria
 - Login/Register Feature
 - Any new user to the website should be able to create an account, which will be stored in a database. Any time the user returns to the website, he/she should be able to log back in to the same account

- Requirements
 1. Newsfeed presentation (front-end)
 - a. Functional:
 - Title: Display aggregated articles and possibly posts
 - Main success scenario: Links and posts are displayed in a way that looks like Facebook, with an infinite scroll
 - Alternative: run out of article links to show user, and show them...a message? A symbol? An animation? That’s a design decision to be made at a later date.
 - Alternative: No friends to show posts from. Say “add friends to see posts!”
 - Alternative: posts are too much, so we just stick to articles
 - b. Non-functional:
 - Newsfeed should load within 2 seconds of user logging in
 - We store article links in a database
 - Front end built using HTML, CSS, and Bootstrap
 2. Newsfeed articles
 - a. Functional:
 - Title: Select untrustworthy or “fake” news sources
 - Main Success Scenario:
 - 1. User scrolls through news feed
 - 2. User clicks on an article

- 3. User is redirected to an external news source that is clearly untrustworthy
- b. Non-functional:
 - Upon loading the page, our API randomly selects untrustworthy news articles and places them into an HTML/bootstrap card.
 - New articles are displayed as the user scrolls down? Or a finite number is generated upon loading the page? This is a design decision we have not made yet.
 - Maybe use politifact.com's section of fact-checked sentences that were declared false?
 - Ideas:
 - <https://github.com/several27/FakeNewsCorpus>
 - <https://github.com/fake-news-detector/fake-news-detector>

3. User Profile

- a. Functional:
 - Title: View user profile
 - Main Success Scenario:
 - 1. User clicks on the "my profile" button in the navigation bar
 - 2. User can see his/her profile picture, name, and birthday.
 - 3. If wanted, the user can update his/her information, including changing the profile picture, name, email, password, or birthday.
- b. Non-functional:
 - Displays profile picture and name above a user's bio
 - Another div lists additional information like their birthday and email address
 - NodeJS retrieves profile information from SQL server and populates HTML fields.

4. Search Bar

- a. Functional:
 - Title: Search
 - Main Success Scenario:
 - 1. User enters one or more strings of characters
 - 2. Parses article titles within database using these "keywords"
 - 3. Returns a list of the top 10 closes matches, based on how many words matched with the article's title
- b. Non-functional:
 - Lists top 10 results which match up with searched keywords
 - Results are displayed as normal articles would be in the newsfeed

- Search results load in 3 seconds

5. Register an Account

a. Functional:

- Title: Register an account
- Main Success Scenario:
 - 1. User loads into website for the first time
 - 2. User clicks login
 - 3. User selects button to register account for new users
 - 4. User fills out registration form
 - 5. User clicks “register”
 - 6. User enters the website logged into their new account
- Alternative: incomplete form
 - User skipped a mandatory field in the registration form or input an incorrect data type (example: not an email in the email input field) and will be requested to re-complete that field. -> return to step 4
- Alternative: Account already exists
 - User attempted to create an account with a username or email that already exists in the system, and he/she will be requested to log on with existing credentials or enter new data. ->return to step 4

b. Non-functional:

- Name Field accepts text
- Password field accepts any character, but is concealed from view
- Confirm password field accepts password type, and must match password
- Email field accepts text with an [xyz]@[xyz].[xyz] type.
- Birthday Field accepts date type
- “Register” button submits form to a SQL server
- SQL server adds a row with the user’s data to the “users” table
- Each input field belongs in a column in the “users” table

6. Log into account

a. Functional:

- Title: Display aggregated articles and possibly posts
- Main Success Scenario:
 - 1. Returning user loads website
 - 2. The user clicks “login”
 - 3. The user enters the correct login credentials

- 4. The user is given access to the website, and his/her profile picture is displayed in the top navbar next to “my profile” button.
- Alternative: User does not exist
 - The user inputted a typo, and must re-complete the login field
 - The user does not actually exist and must click “register” to register a new account
- b. Non-functional:
 - Name Field accepts text
 - Password accepts characters and hides them when typed
 - Login button takes you to the home page (newsfeed)
 - Login button retrieves profile information from SQL server and populates the “profile” page with the correct data with nodeJS.
- Project Plan
 - We have a project plan board on Jira.com
 - It is an Kanban-style agile project board where we can link tasks and have subtasks.
 - For each feature, we have a designing, coding, testing, and integrating task on the agile board. Each of these can be marked as “to do”, “in progress” or “done”.
 - Each task is sorted by a priority, can be assigned individual contributors, and belong to a sprint
 - Every sprint has a logical order of how the application should be developed, and all future sprints rely on previous ones being completed.

Milestone Two – Project Plan

Project Features list	<ul style="list-style-type: none">• Document (as a bullet point list) the major FEATURES of your software application.• Each features should have a TITLE followed by a brief DESCRIPTION of that feature.
Requirements	<ul style="list-style-type: none">• Document the key FUNCTIONAL and NON-FUNCTIONAL requirements for a minimum of SIX features. Follow one of the formats/templates provided in class. Your application will likely have many more features, and you should create requirements documents for all the features. But for this milestone, you need only turn in SIX requirements documents.
Project Plan	<ul style="list-style-type: none">• Create a Project Plan in your chosen Project Management tool. May be a GANTT chart, Kanban board, or other suitable format.• The Project Plan should include each Feature that is to be developed.• The Project Plan should indicate the SEQUENCE of Sprints. That is, the plan should show in what order the features will be developed (designed, coded, tested, integrated.)• The Project Plan should show for each feature WHEN (by calendar date) the feature will be worked on.• The Project Plan should show for each feature WHO will be working on the development of the feature.• The Project Plan should result in a completed, working application that is ready to present by April 26.