

This project is a web platform designed to aggregate articles from multiple sources, encourage meaningful discussions, and provide ratings to evaluate article quality and trustworthiness. It aims to create a better reading experience by integrating user feedback and verified insights.

#### Minimum Viable Product

- Open the website
- Browsing section with headlines of articles screen scraped from 3 or 4 different sources
- Get at least 5 from every source, update every 4 hours
- On the Browse Articles page, show average total rating for each article
- Pop up button on average rating icon showing averages for each category (bias, accuracy, quality)
- Verified user login in the top right
- Open an article
- See verified user comments on the article with their ratings for each category and a brief explanation
- Can like or dislike the verified user comment
- See whole article
- General comments below
- Upvote / Downvote general user comments (-5 votes deleted)
- If you login as a verified user, basically the same but there is an extra button to add verified users comment

## Epic 1: User Interface and Navigation

#### User Stories:

1. As a user, I want to open the website and access a simple and intuitive homepage so that I can quickly understand the layout and find what I need.
2. As a user, I want a browsing section that lists headlines from multiple sources so that I can easily explore different articles.
3. As a user, I want to access and read entire articles in a clean, readable format so that I can focus on the content without distractions.
4. As a user, I want to see general comments and engage with discussions on articles so that I can participate in conversations and understand diverse perspectives.
5. As a programmer of VeriTag, I want to have manually loaded articles as a placeholder for the screen scraper so that I can best design the data in the UI.

## Epic 2: Article Data Aggregation and Presentation

#### User Stories:

1. As a user, I want to see at least 5 articles from each of 3–4 sources, updated every 4 hours so that I have access to fresh and diverse content regularly.
2. As a user, I want to see the average total rating for each article on the Browse Articles page so that I can quickly gauge its overall quality.
3. As a user, I want to view detailed category ratings (bias, accuracy, quality) by clicking a pop-up button on the average rating icon so that I can better understand how the article was evaluated.

## **Epic 3: User Authentication and Interaction**

### **User Stories:**

1. As a verified user, I want to log in securely and see my status reflected on the website so that I can access additional features and trust my data is safe.
2. As a general user, I want the ability to post general comments on articles and vote on other users' comments so that I can participate in discussions and share my opinions.
3. As a verified user, I want an additional button to add my own verified comments on articles with ratings for each category and a brief explanation so that I can provide credible and constructive feedback.

## **Epic 4: User Feedback and Engagement**

### **User Stories:**

1. As a user, I want to like or dislike verified user comments so that I can signal their usefulness to other readers.
2. As a user, I want to upvote or downvote general user comments, with those receiving -5 votes automatically deleted so that the quality of discussions is maintained.
3. As a verified user, I want my comments to stand out as trustworthy and informative to other readers so that I can build credibility and add value to discussions.
4. As a user, I want to ensure that article feedback (comments, ratings) contributes to building a helpful reading experience so that the platform continuously improves and meets users' needs.

## DELIVERABLES

In addition to the prioritized requirements in your Trello board, you will submit a short design document describing your initial system design/architecture. Include sections for:

### 1. System Description

- a. Explain the purpose of the system (in your own words)
- b. Give a basic overview of the functionality provided by this system

### 2. Architectural Overview

- a. describe the design/architecture of your system
- b. include a list of all major system components
- c. include a system diagram

### 3. Functional Requirements

- a. List at least four epics that describe what you believe are the most important large features that are needed for this project

- i. Recall that epics are user stories that are too large to be completed in a single sprint. Epics are typically split into many small stories. • Don't forget to use Trello labels to associate user stories with related epics.

- ii. For example, a few epics for an e-commerce application might be:

EPIC 1 — Shopping History

As an online shopper

I want to view the history of my previous purchases

So that I can reorder items that I previously liked

EPIC 2 — Review Purchases

As an online shopper

I want to see reviews for the items I'm considering buying

So that I can make more informed decisions about the items I purchase

### 4. Non-Functional Requirements

- a. Depending on the project, you may have non-functional requirements. Document those here as well.
- b. Note: non-functional requirements are not typically written in user story format. Example:

Non-Functional Req. 1 — System Response Time

The service's runtime environment must be secure against the execution of malicious code.

5. Technologies and Frameworks

- a. list and describe the programming languages, frameworks, databases, etc. that you plan to use

6. Define your Minimum Viable Product (MVP)

- a. Your MVP can simply be a paragraph that describes what you believe must be minimally achieved for your product to be useful for its intended audience.
- b. Referring to relevant user stories in your Kanban's backlog is a plus. This will make your MVP more tangible and help guide later development efforts.

7. Create a Preliminary Roadmap for the Semester

- a. Create a preliminary timeline for when you would like to have high-level features implemented and your MVP completed.
- b. There are six development sprints with pre-defined dates. Use these to structure your roadmap.

## Technologies Used

- **Frontend:**
- **Backend:**
- **Database:**
- **Web Scraping:**
-