

# Slop Mop - Sprint 1 Planning

**Team 7** : Aryan Rakshit, Dhiya Pereira, Jack White, Seung Hyeon (Leah) Lee,  
Raj Penmetcha, Seth DeWhitt

## Sprint Overview (0.5 points)

(a) Discuss the overview of this sprint (2/16 - 3/6)

- Establish baseline infrastructure for the AI detection system by developing a basic text detection mode, setting up the backend API and creating an MVP extension (add more here)

(b) Clearly state who the SCRUM master (team leader) is and briefly mention your scrum meeting schedules.

- Seth DeWhitt
- Our meetings are Wednesdays 2-2:30pm, LWSN 1190 with TA Zhou Xhan.

(c) Include risks and challenges for this sprint, if you have any.

- Model Accuracy
  - Achieving a high accuracy for detection may require multiple iterations, testing out different models, or fine tuning
- Extension compatibility across different browsers
- API Response
  - Meeting our 10 second detection requirement while handling model inference
- Team Coordination
  - Learning to work together effectively with rest of team
- Learning Curve for different tools
  - Some team members may need to get more familiar with tools such as PyTorch, FastAPI, and possibly different transformer models
- Web scraping in-situ for diverse social media sites.

## Current Sprint Detail (4.0 points)

(a) List all user stories to be implemented in this sprint.

- (b) For each user story, add a list of well-defined, self-contained tasks.
- (c) Ensure to include “testing” or “unit tests” tasks for each appropriate user story.
- (d) Add a description for each task, and clearly state which team member is assigned to the task and its workload estimation (in work hours - make sure to distribute the total workload evenly among team members!). Task description should be clear. All tasks that will be Vibe coded must be clearly labeled as such.
- (e) Add THREE or more detailed acceptance criteria which defines the set of conditions or statements in order for a user story to be accepted. Using “Given (some precondition) When (I do some action) Then (I expect some result).” format is strongly recommended. Typical number of successful, good acceptance criteria is about five.

1. 4: “As a potential user, I would like a smooth and intuitive website design, with clear and concise instructions regarding installation.”
  - a. Workload Estimation: ~20 total hours
    - i. Mostly from general website setup
  - b. Assigned: Seth DeWhitt & Raj Penmetcha
  - c. Tasks:
    - i. Initialize the website
      1. **Assigned to:** Seth DeWhitt
      2. **Workload:** 3 hours
      3. **Description:** Set up Next.js. Configure the project structure with necessary folders such as components, utils and styles. Initialize the Git repository and setup ESLint and Prettier for code quality and formatting
    - ii. Design homepage with instructions to install
      1. **Assigned to:** Raj Penmetcha
      2. **Workload:** 4 hours
      3. **Description:** Create a homepage with a section that explains the purpose and benefits of the extension. Include step by step instructions to download the extension and any necessary features. Design should be clear and responsive.
    - iii. Initialize FAQ page and navigation bar
      1. **Assigned to:** Seth DeWhitt
      2. **Workload:** 4 hours
      3. **Description:** Create a navigation bar with links to Home, FAQ, Privacy Policy and possibly intention/story. The footer can have contact information and legal links. The FAQ page should have 5-6 common questions about accuracy, privacy, platforms and possibly troubleshooting installation
    - iv. Add basic authentication so user data can be stored

1. **Assigned to:** Raj Penmetcha
2. **Workload:** 4 hours
3. **Description:** Implement user authentication (possibly with firebase auth or similar). Create login with email/password and Google OAuth option
- v. Add user settings page with preferences
  1. **Assigned to:** Seth DeWhitt
  2. **Workload:** 4 hours
  3. **Description:** Build a setting page where users can manage account details and add disabled websites. In future sprints, planning to add preferences such as dark mode, notifications, language, etc. Possibly include an account deletion option and make sure to save preferences to the backend.
- vi. Set up CI/CD pipeline with platforms like Jenkins or GitHub Actions
  1. **Assigned to:** Raj Penmetcha
  2. **Workload:** 3 hours
  3. **Description:** Configure GitHub actions for automation and deployment to Vercel. Set up automatic deployments on a merge to main branch. Add environment variables. Document process in readme.
- d. Acceptance Criteria
  - i. *Given* a first time visitor going to the homepage *when* they scroll through the page *then* they see a clear explanation of what the extension does, why it is useful and the step by step instructions that are easy to follow
  - ii. *Given* the user reads the instructions *when* they follow the steps provided, *then* they can install the chrome extension within a few minutes without needed to ask for additional help
  - iii. *Given* a user wants to create an account *when* they click “sign up” and complete the create account form or Google OAuth *then* their account is created successfully and are redirected to settings page or homepage
  - iv. *Given* a user has a question about the extension *when* they navigate to the FAQ page via the navigation bar *then* they find clearly written questions covering common topics with short and clear answers
  - v. *Given* a developer pushes code to the main branch *when* the github workflow executes *then* the code is automatically deployed to vercel.

- e. Unit Tests
  - i. Home Page
    - 1. renders a clear purpose/benefits section
    - 2. renders installation instructions as ordered steps
    - 3. Has download/install button/link and download succeeds.
  - ii. Navigation
    - 1. Renders working links to Home, FAQ, Privacy Policy, About
  - iii. FAQ Page
    - 1. Renders at least 5-6 FAQ items
    - 2. Includes expected topics like accuracy, privacy, supported platforms, installation, troubleshooting, simple/accessibility mode.
  - iv. Authorization
    - 1. Email/password signup possible.
    - 2. Verify that login is possible with a new account.
  - v. Settings
    - 1. Loads user settings on webpage loading. Assert fields are populated
    - 2. Updates disabled sites list.
    - 3. Saves preferences
- 2. 12: “As a general user, I would like to disable detection on specific websites so I can control where the extension runs.”
  - a. Workload Estimation: ~9 hours
  - b. Assigned: Seth DeWhitt & Raj Penmetcha
  - c. Tasks:
    - i. Design and implement extension settings page UI
      - 1. **Assigned to:** Seth Dewhitt
      - 2. **Workload:** 5
      - 3. **Description:** Create a popup settings page (or separate options page on website). Design UI section to manage disabled websites with a field to add urls, list of already disabled websites and delete/edit buttons for each entry.
    - ii. Implement URL pattern matching
      - 1. **Assigned to:** Raj Penmetcha
      - 2. **Workload:** 5
      - 3. **Description:** Create a URL matching system to check if the current website is in the disabled list. Make sure it supports “domain-level matching” meaning that it will disable all subdomains. (\*.reddit.com” disables all reddit domains).
  - d. Acceptance Criteria

- i. *Given* a user opens the extension setting page *when* they enter a website in the input field and click add, *then* that website appears in the disabled websites list and is saved to the extension storage immediately
- ii. *Given* a user has a website in their disabled websites list, *when* they visit that website, *then* no AI detection runs on the page, including all of its subdomains
- iii. *Given* a user viewing their disabled websites list *when* they click the delete button next to a website, *then* the website is removed from the list and detection resumes on the now deleted site
- iv. *Given* a user enters an invalid URL, *when* they attempt to add it to the disables list *then* an error message appears explaining the URL format is invalid

e. Unit Tests

- i. Test URL pattern matching function
    - 1. assume "[reddit.com](https://reddit.com)" is blacklisted, then:
      - a. "<https://reddit.com/r/test>" is blocked
      - b. "<https://reddit.com/>" is blocked
      - c. but "<https://notreddit.com/r/test>" passes
    - 2. Wildcard form disables subdomains. Assume "[\\*.reddit.com](https://reddit.com)" is blacklisted, then:
      - a. "[www.test.reddit.com](https://www.test.reddit.com)" is blocked
      - b. "[reddit.com](https://reddit.com)" follows intended behavior
    - 3. Invalid inputs handled safely. "not a url" is false or throws error.
  - ii. Extension Settings Page
    - 1. Initial load populates list of settings
    - 2. Test adding a site to blacklist. Assert it appears immediately and array of blacklisted sites is updated in code.
      - a. Assert no duplicates exist in array
    - 3. Assert deletion of a site removes it from the blacklist array in memory
    - 4. Assert the blacklist is saved to disk.
3. 14: "As a general user, I would like the system to be accurate at detecting AI-generated content so that I am not being misled by the extension itself."
- a. Workload Estimation: ~42 hours (text) ~80 hours (image but will be implemented next sprint)
  - b. Assigned: Seung Hyeon (Leah) Lee & Seth DeWhitt
  - c. Tasks:
    - i. Text Model

1. Initialize pre-trained Transformer architecture using HuggingFace (write PyTorch code)
  - a. **Assigned:** Seung Hyeon (Leah) Lee
  - b. **Workload estimation:** ~2 hours
2. Develop preprocessing pipeline using HuggingFace and custom text cleaning rules like remove emoji, url or any other things that can confuse the model
  - a. **Assigned:** Seung Hyeon (Leah) Lee
  - b. **Workload estimation:** ~6 hours
3. Write training script to adapt the model to social media data
  - a. **Assigned:** Seung Hyeon (Leah) Lee
  - b. **Workload estimation:** ~10 hours
  - c. Details:
    - i. Training loop, optimizer
    - ii. Integrate TensorBoard/Weights&Biases to help visualize the loss curve
4. Prepare and rigorously test the model
  - a. **Assigned:** Seung Hyeon (Leah) Lee
  - b. **Workload estimation:** ~12 hours
  - c. Details:
    - i. Implement automated evaluation script (evaluate.py)
    - ii. Write script that will programmatically inject noise into test data and re-run model to measure stability (stress-test tool)
    - iii. Write code that will generate precision-recall curve and mathematically determine the optimal confidence threshold to increase accuracy of the model
5. Write code to separate testing/training dataset to train (70-80%), validation (10-15%) and test (10-15%) (~2 hours)
  - a. **Assigned:** Seth DeWhitt
  - b. **Workload estimation:** 2 hours
- ii. Export trained model
  1. **Assigned:** Seth DeWhitt
  2. **Workload estimation:** 8 hours
- d. Acceptance Criteria:
  - i. Given a clear AI-generated text sample
    1. When detection is triggered
    2. Then the system returns a confidence score above a 70 %.

- ii. Given a clearly human sample
    - 1. When detection is triggered
    - 2. Then the system returns a confidence score below 30 %.
  - iii. Given a mixed AI-generated and human text
    - 1. When detection is triggered
    - 2. Then the confidence score shows uncertainty (between 40-60%).
  - iv. Given the model evaluation testing
    - 1. When evaluated on a benchmark dataset
    - 2. Then the model accuracy must meet at least beeline threshold, like 70%.
  - v. Given a post is flagged as AI-generated
    - 1. When results are displayed
    - 2. Then the explanation and confidence score align logically with the model output.
- e. Unit Test
  - i. The model correctly detects AI generated, mixed and human posts with the accuracy of 80%
- 4. 17: "As a general user, I would like a short explanation of why a post was flagged as AI-generated so I can judge the credibility of the result."
  - a. Workload Estimation: ~ 5 hours each
  - b. Assigned: Aryan Rakshit & Dhiya Pereira
  - c. Tasks:
    - i. Develop logic that generates a short, user friendly explanation based on confidence score and detected patterns such as repetition, structure consistency, or stylistic markers.
    - ii. Ensure the explanation is limited to 2-3 concise sentences, avoids technical jargon, and maintains a consistent tone.
    - iii. Display the generated explanation directly below the confidence score for both manual and automatic detections
    - iv. Adjust wording to reflect uncertainty for mid-range confidence scores and handle cases where detection metadata is incomplete
    - v. Tests with AI, human, and mixed samples to verify explanations are accurate, aligned with confidence levels, and easy to understand.
  - d. Acceptance Criteria
    - i. Given the content is flagged as AI-generated
      - 1. When the result is displayed
      - 2. Then a short explanation is shown below the confidence score.
    - ii. Given explanation is generated

1. When the user reads it
    2. Then it references patterns such as repetitive phrasing, structure consistency, and stylistic markers.
  - iii. Given confidence is low
    1. When result is displayed
    2. Then the explanation reflects uncertainty.
  - iv. Given explanation text is long
    1. When displayed
    2. Then it is limited to 2-3 concise sentences.
  - v. Given a post is flagged as AI-generated
    1. When the explanation is shown
    2. Then it must be understandable by non-technical users.
  - e. Unit Tests
    - i. Verify that for each scanned and classified post:
      1. A badge appears with an explanation limited to 1-3 concise sentences, shorter is better.
5. 15: “As a general user, I would like to see a confidence percentage which indicates how likely a post is AI-generated so I can better judge credibility” & 18: “As a general user, I would like confidence ranges or uncertainty indicators so that I know when results are less reliable.”
- a. Workload Estimation: 20 hours
  - b. Assigned: **Raj Penmetcha** & **Aryan Rakshit**
  - c. Tasks:
    - i. Be able to calculate confidence (float) in DetectorModel (either text or image)
      1. **Assigned to:** **Raj Penmetcha**
      2. **Workload:** 6 hours
      3. **Description:** Implement confidence score extraction from model output (softmax prob). Should work for both text and image detection models
    - ii. Decide on calculation logic
      1. **Assigned to:** **Raj Penmetcha**
      2. **Workload:** 3 hours
      3. **Description:** Define confidence ranges, research optimal threshold values based on model performance
        - a. High confidence  $\geq 70\%$
        - b. Uncertain: 40-60%,
        - c. High confidence human:  $\leq 30\%$
    - iii. Display confidence score/percentage on screen to user



1. **Assigned to:** Raj Penmetcha
  2. **Workload:** 5 hours
  3. **Description:** Design UI badges showing percentage and implement color coding (red for high AI likelihood, yellow for uncertain, green for human). Add things like visual cues (icons, progress bars) for confidence ranges. Also make sure major design changes are responsive to different screen sizes
- iv. Test: feed multiple posts with varying ai-contribution
1. **Assigned to:** Aryan Rakshit
  2. **Workload:** 5 hours
  3. **Description:** Collect test dataset with known AI/human content at various levels. Validate that the confidence scores align with actual content origin. Also measure calibration meaning if 70% confidence = 70% accuracy/.
- d. Acceptance Criteria
- i. Given detection is successfully completed
    1. When results are displayed
    2. Then a confidence percentage (0-100%) is clearly shown to the user.
  - ii. Given the model produces a prediction score
    1. When the confidence value is calculated
    2. Then it must be converted into a float percentage between 0% and 100%.
  - iii. Given the confidence score is above the defined high-confidence threshold ( $\geq 70\%$ )
    1. When the result is displayed
    2. Then the UI clearly indicates a high likelihood of AI-generated content.
  - iv. Given the confidence score falls within the defined uncertainty range (40% - 60%)
    1. When the result is displayed
    2. Then the UI indicates that the result is uncertain or mixed.
  - v. Given the confidence score is below the defined low-confidence threshold ( $\leq 30\%$ )
    1. When the result is displayed
    2. Then the UI indicates a high likelihood of human-written content.
  - vi. Given a confidence percentage is displayed
    1. When an explanation is shown (Story 17)

2. Then the explanation wording must logically align with the confidence range
- vii. Given detection is triggered either automatically or manually
  1. When results are displayed
  2. Then the confidence percentage and range indicators must behave consistently in both cases.
- e. Unit Test
  - i. Verify that for each scanned and classified post:
    1. A badge appears with a likelihood rating: high, low, uncertain, or exact percentage. Percentages must be between 0 and 100%.
6. 21: "As a user installing the extension for an older generation, I wish the interface to be age-inclusive or have a setting to enable (turn on) age-inclusive, low vision accessible interface (control font size)."
- a. Workload Estimation: 15 hours
- b. Assigned: [Dhiya Pereira](#)
- c. Tasks:
  - i. Change the font size for all pages and thus modify the UI so that it still looks good and is not that different from the normal display
  - ii. Change buttons with only icons to include short text to describe its function or make it simple and more straight-forward
  - iii. Add a button or a toggle to settings page so that the user can turn it on and off
- d. Acceptance Criteria
  - i. Given the settings page has been opened, an accessible vision option should be easily visible and triggerable. Then the vision changes occur
  - ii. Given vision accessibility mode is enabled, font should be easily visible from a distance in high contrast colors.
  - iii. Given simple mode is enabled, advanced options should disappear and a simple extension on/off button can be triggered.
- e. Unit Test
  - i. Verify that simple mode removes all 'advanced' settings and a simple 'On/Off' button appears.
  - ii. Verify that there are no display/rendering issues when simple, accessible, and dark/light mode are enabled in combination.
  - iii. Verify that accessibility mode changes font size and contrast to high visibility versions.
7. 24: "As a general user, I would like to manually trigger detection on a specific post so that I can recheck content."

- a. Workload Estimation: ~ 10 hours
  - b. Assigned: [Aryan Rakshit](#)
  - c. Tasks:
    - i. Initialize backend if not done yet
      - 1. Investigate use of open source systems for AI-detection
    - ii. Design and Implement "Detect Now" Button UI
    - iii. Implement Post Content Extraction Logic
    - iv. Connect Frontend to Backend Detection API
    - v. Add Loading State & UX Feedback
    - vi. Implement Error Handling & Retry Logic
    - vii. Unit Testing & Edge Case Testing
  - d. Acceptance Criteria
    - i. Given the extension is active on a supported webpage
      - 1. When the page finishes loading
      - 2. Then a "Detect Now" button is visible for the current post.
    - ii. Given a post is visible on the screen
      - 1. When I click the "Detect Now" button
      - 2. Then the system runs detection on that specific post and displays the result.
    - iii. Given multiple posts are visible on the page
      - 1. When I click "Detect Now" on one specific post
      - 2. Then only the selected post is analyzed
    - iv. Given I click the "Detect Now" button
      - 1. When detection is in progress
      - 2. Then a loading indicator is displayed until results are returned
    - v. Given I click "Detect Now"
      - 1. When detection fails due to a network or server issue
      - 2. Then an appropriate error message is displayed and the user can retry.
  - e. Unit Test
    - i. After clicking Detect Now, verify on the server side that a new request was sent, the request was not retrieved from cache, and was analyzed and sent to the extension. We can do this via logging.
8. 29: "As a general user, I would like the extension to also work in private/incognito mode so that I can browse securely. "
- a. Workload Estimation: 5 hour
  - b. Assigned: [Seth DeWhitt](#)
  - c. Tasks:

- i. Research incognito requirements
    - ii. Match chromium's requirements for incognito mode
    - iii. Add a check in local and server caching for incognito mode
      - 1. Prevent caching when incognito == True
    - iv. Inform the user with a popup that their data is sent to a central server for processing.
  - d. Acceptance Criteria:
    - i. Given a Reddit/X/LinkedIn,etc post viewed in incognito Mode, the extension should successfully read the content, classify it and display the result to the user.
    - ii. Given a post has been scanned in incognito, that post should not appear in the server or local cache
    - iii. Given the extension is running in incognito, a warning message should appear to the user informing them that data is sent to a central server for inference
  - e. Unit Test
    - i. Extension runs on incognito and successfully classifies posts and displays them. No cache history is saved or visible. User receives privacy notice upon opening incognito.
9. 30: "As a general user, I would like the extension to support dark mode on my computer/mobile device so that it integrates visually with my browser settings."
- a. Workload Estimation: ~2 hours
  - b. Assigned: Seung Hyeon (Leah) Lee
  - c. Tasks:
    - i. Detect display/theme mode of the user device and change the display to fit the user's display/theme mode.
    - ii. Add a separate toggle or a button to settings page so that the user can also manually change to dark/light mode.
  - d. Acceptance Criteria
    - i. Given user's display theme is light mode
      - 1. When the user opens the extension popup
      - 2. Then the UI will also be in 'light mode' and integrate flawlessly
    - ii. Given user's display theme is dark mode
      - 1. When the user opens the extension popup
      - 2. Then the UI will also be in dark mode' and integrate flawlessly
    - iii. Given user's display theme is light mode
      - 1. When the user changes the display theme from light to dark
      - 2. Then the UI seamlessly transitions from light to dark

- iv. Given user's display theme is dark mode
  - 1. When the user changes the display theme from dark to light
  - 2. Then the UI seamlessly transitions from dark to light
- v. Given a dark/light mode toggle exists in the settings page
  - 1. When the user clicks on the toggle
  - 2. Then the toggle moves from one side to the other smoothly

e. Unit Test

- i. If dark mode is selected, all background is black and all font is white.
- ii. If light mode is selected, all background is white and all font is black.
- iii. If 'Match Browser' mode is selected, the extension seamlessly follows the browser's lead and matches the website.

10.34: "As a general user, I would like the extension to show an error message when a detection fails."

a. Workload Estimation: ~ 8 hours

b. Assigned: [Aryan Rakshit](#)

c. Tasks:

- i. Pop-up with different error messages for different cases (time-out, network etc)
- ii. Suggest ways to fix this issue to the user if applicable
- iii. If the detection still fails after retrying automatically up to a limit (user story 35), display a different error message. Maybe a link to Q&A page or report button(if report function is not implemented, just have a button that will still link to report popup/tab but the report itself doesn't get sent to the server)

d. Acceptance Criteria:

- i. Given detection is triggered (either automatically or manually)
  - 1. When the backend returns a server error or model failure
  - 2. Then a pop-up error message appears indicating the detection service is temporarily unavailable.
- ii. Given I trigger detection
  - 1. When the detection request fails due to a network or timeout issue
  - 2. Then a clear error message is displayed and the user is given the option to retry.
- iii. Given I am scrolling through a supported feed
  - 1. When the extension automatically triggers detection on a post and the request fails due to a network issue

2. Then an error message is displayed on or near that post indicating a network error..
  - iv. Given detection fails and the system retries automatically up to a predefined limit
    1. When the retry limit is reached without success
    2. Then a final error message is displayed informing the user that multiple attempts were unsuccessful.
  - e. Unit Test
    - i. For 10 posts, when background reaches some erroneous state, this should be reflected in a visible error message to the user clearly visible on the post, such as in a badge.
11. 39: "As a general user, I would like the extension to indicate when a language is not supported."
- a. Workload Estimation: ~ 2 hours
  - b. Assigned: [Aryan Rakshit](#)
  - c. Tasks:
    - i. Add language verification before either responses are returned or the content is added to the pipeline
  - d. Acceptance Criteria
    - i. Given a post is loaded on a supported website
      1. When detection is triggered
      2. Then the system verifies the language before running the AI detection model
    - ii. Given the detected language is not supported by the system
      1. When detection is triggered
      2. Then the extension displays a clear message stating that the language is not supported.
    - iii. Given the language is unsupported
      1. When detection is triggered
      2. Then the AI detection model does not run and no confidence score is generated.
    - iv. Given an unsupported language message is shown
      1. When the user views the message
      2. Then it clearly explains which languages are currently supported
    - v. Given the detected language is supported
      1. When detection is triggered
      2. Then the system proceeds normally and displays a confidence score and explanation.
  - e. Unit Test

- i. When showing a post on a supported site but the language is not English or Spanish, the extension should provide a result but in the badge indicate that the language is not supported and the result is inaccurate.

12.41: “As a general user, I would like the extension to display a disclaimer reminding that detection results are based on probability.”

- a. Workload Estimation: 10 hours
- b. Assigned: [Dhiya Pereira](#)
- c. Tasks:
  - i. Add a disclaimer to website and on the extension pop-up (if extension has been implemented and it works)
- d. Acceptance Criteria
  - i. Given a detection result is displayed
    - 1. When the user views the confidence score
    - 2. Then a disclaimer must appear stating that the result is probabilistic and not a definitive determination.
  - ii. Given the extension popup is opened after a detection
    - 1. When the results are shown
    - 2. Then the disclaimer must be visible without requiring the user to scroll/click for more additional information.
  - iii. Given the user visits the project website
    - 1. When detection capabilities are described
    - 2. Then a disclaimer must clearly state that AI detection results are based on probability and may not be 100% accurate.
- e. Unit Test
  - i. Opening the extension for the first time opens up a disclaimer page.
  - ii. Opening the popup settings window displays a disclaimer in the corner.

13.49: “As a Reddit user, I would like for the extension to work on Reddit so that I can detect AI content on Reddit.”

- a. Workload Estimation: 30 hours
- b. Assigned: [Jack White](#)
- c. Tasks:
  - i. Develop PostExtractor class
    - 1. Define MVP output and scope.
    - 2. Identify target surfaces to scrape
    - 3. Determine DOM architecture and develop strategy.
      - a. Research Reddit scraping techniques
    - 4. Implement getStablePostID(postNode) rules
    - 5. Implement Text Extraction pipeline

6. Implement Image Extraction pipeline
    - a. Classify content as image/video/text/image
  7. Implement performance optimizations
  8. Unit tests and manual validation. Temporarily log extracted fields, extraction timing, all content of NormalizedPostContent
- ii. Develop OverlayRenderer class
    1. Define UI requirements
    2. Identify target containers / anchor points to inject into on Reddit
    3. Implement safe DOM injection
    4. CSS and Theming to support accessibility, dark/light backgrounds,
    5. Implement state management and update logic
    6. Handle dynamic updates / reloading of posts during infinite scroll
    7. Handle performance concerns, create performance criteria
    8. Validate results
- d. Acceptance Criteria
    - i. Given text and image reddit posts, the PostExtractor should return NormalizedPostContent objects to the FeedObserver class. These objects should contain little to no fluff/extra content
    - ii. Given text comments, the PostExtractor should extract the top 25 comments at a depth of 1 (each comment is the root of a tree)
    - iii. Given a reddit post and a DetectionResponse from the backend, the OverlayRenderer should display a badge and confidence percentage in the top right of the post.
  - e. Unit Test
    - i. Given a Reddit text post or text comment or combination text/image post, the extension should successfully populate a NormalizedPostContent object and send to the FeedObserver class. The object should contain just the text and image content of the post and no fluff or artifacts.
    - ii. Given a DetectionResponse object and a corresponding Reddit post is visible on the screen, a badge should appear representing the classification returned in the DetectionResponse. It should be visually similar to the UI mockup in our Design Document.



## Backlog (0.2 points)

(a) Include all the other user stories from your Product Backlog document.

1. As a general user, I would like to scroll through my social media feed quickly without noticeable performance drops due to the extension so that the extension serves as an enhancement with no drawbacks.
2. As a general user, I would like detection results to update automatically as I scroll so that newly loaded content is analyzed in real time.
3. As a general user, I would like detection indicators to be visually distinct but non-intrusive so that they do not obstruct my ability to read posts.
4. As a potential user with doubts regarding the extension, I would like for there to be an FAQ page so that I can likely answer my questions without needing to reach out to customer support.
5. As a general user, I would like in-extension onboarding tips so that I can understand how detection results are presented.
6. As a general user, I would like to hover over a detection indicator to view more detailed information so that I can understand results without leaving the page.
7. As a general user, I would like to have recent (last 24 hours) detections cached so that I will not have to wait for the extension to re-detect when viewing a post again.
8. As a general user, I would like to view a history of the cached analyzed posts so that I can revisit past detection results and sites where they were detected.
9. As a general user, I would like the extension's detection and analysis functionality to be consistent across different social media platforms so that I have a unified experience.
10. As a general user, I would like for there to be user statistics so that I can visibly observe the effectiveness of the extension.
11. As a general user, I would like the system to be able to ingest video content so that I can detect AI use in videos.
12. As a general user, I would like to report any incorrect detections so that the system can improve over time.
13. As a general user, I would like to access verified/reliable supporting materials for non-flagged content so that if I want to I can further verify it myself.
14. As a system, I would like to apply rate limiting per user or per session so the backend services are not overwhelmed.
15. As a general user, I would like the extension to detect AI-generated content in comments and replies so that online discussions can also be evaluated.

16. As a general user, I would like detection to happen on sponsored posts and advertisements so that promotional content is also evaluated.
17. As a general user, I would like to be able to collapse detection indicators when on a webpage, so that the information does not clutter the screen.
18. As a general user, I would like to receive a notification of detection when it has finished for content I may have scrolled past so that I do not miss results.
19. As a general user, I would like the extension to distinguish between AI-generated, AI-assisted, and human-written content for more specific results.
20. As a general user, I would like the extension to pause detection automatically on low battery mode so that device resources are conserved.
21. As a general user I would like the extension to avoid reanalyzing the same content multiple times in the same session.
22. As a general user, I would like the extension to prioritize posts that are only currently visible on the screen.
23. As a general user, I would like the extension to retry failed detections automatically up to a limit.
24. As a general user, I would like to enable or disable text detection and image detection independently so I can customize what is analyzed.
25. As a general user, I would like the extension to remember preferences such as enabling or disabling certain websites, dark/light mode, and turning off certain websites even when the browser is restarted.
26. As a general user, I would like the extension to detect the language of a post so the behaviour of detection can adapt.
27. As a general user, I would like the extension to avoid flagging content that is labeled satire or parody so nothing is misclassified.
28. As a general user, I would like to search for detection history by keyword.
29. As a general user, I would like the extension to highlight specific segments of the text that triggered the AI detection.
30. As a general user, I would like the extension to have a "Manage Disabled Websites" dashboard/settings so that I can easily view and manually add or remove websites from my exclusion list.
31. As a user with old devices, I would like the extension to automatically pause when I am browsing on a different tab or an application to conserve CPU and RAM.
32. As a general user, I would like to have "Scan entire page" to force the extension to analyze every visible post at once if the automatic scroll detection fails.
33. As a Google user, I would like the extension to apply to AI-made webpage descriptions, articles, and the "AI Overview".
34. As a LinkedIn user, I would like for the extension to work on LinkedIn so that I can detect AI content on LinkedIn.

- 35. As a X/Twitter user, I would like for the extension to work on X/Twitter so that I can detect AI content on X/Twitter.
- 36. As a Facebook user, I would like for the extension to work on Facebook so that I can detect AI content on Facebook.
- 37. As an Instagram user, I would like for the extension to work on Instagram so that I can detect AI content on Instagram.
- 38. As a general user, I would like for the extension to have fact-checking capabilities so that I can verify if social media content is accurate, whether it is AI-generated or not.

(b) (Optional, will not be graded) List tasks for each user story.