Plot Armor is a browser extension designed to block spoilers from websites and social media, ensuring your favorite shows, movies, and events remain unspoiled.

1. Core Components

- Main screen/Overlay/Dashboard
- Settings/Options Page

● Main Screen Breakdown:
  ○ Purpose: Provide users with control over their spoiler settings
  ○ Features:
    ■ Toggle Button (activate/deactivate)
    ■ Recent flagged content overview
    ■ Quick action buttons (e.g. "Whitelist site", "View blocked content")
    ■ Notification Area (e.g. "5 spoilers blocked today!")

● Settings/Options Page Breakdown:
  ○ Purpose: Allow users to customize the behavior of the spoiler blocker
  ○ Features:
    ■ Add keywords, phrases, or topics to block (e.g. "Spider-Man", "Breaking Bad", "death")
      ● Utilize AI/NLP to enhance topic-censoring functionality
    ■ Manage pre-configured spoiler lists for popular shows, movies, or games
    ■ Customize blocking style (e.g. blur entire page, blur only sentences containing the spoiler, show warnings)
    ■ Whitelist/blacklist specific websites
    ■ Dark mode/appearance settings

2. Functional Components

● Keyword Matching and Content Detection System
  ○ Purpose: Identify spoilers in text, images, or videos based on user preferences
  ○ Features:
    ■ Use NLP (Natural Language Processing) to detect related phrases
    ■ Context-based analysis to improve precision (instead of blindly blocking all instances of a keyword)
      ● Toggled off by default (WILL allow users to censor all instances of a keyword no matter the context)

- ■ ==Potential feature (v2):== provide compatibility with social media slang and abbreviations
    - ● This does go beyond the original scope of the extension but is a worthwhile idea
- ● Blocking Mechanism
    - ○ Purpose: Actively hide or blur spoiler content in real-time
    - ○ Features:
        - ■ Blur text (keyword/sentence/phrase/whole page) based on user preference
            - ● For a more subtle option, users can choose a banner to appear for each instance of a spoiler, rather than any blur option
        - ■ Replace spoiler text with a customizable placeholder (e.g. "Spoiler Detected!")
        - ■ ==Potential feature (v3):== blur images related to keywords/key phrases/blacklisted media
        - ■ ==Potential feature (v4):== block videos/suggest skipping sections where spoilers appear
- ● Real-Time Scanning
    - ○ Purpose: Monitor websites dynamically as they load
    - ○ Features:
        - ■ Scan text, images (v3), or videos (v4) in real-time
        - ■ Avoid lagging the browser by optimizing scanning performance
        - ■ Alert users with a banner when spoilers are detected instead of blocking them automatically
            - ● Optional setting
- ● Whitelist and Blacklist Management
    - ○ Purpose: Let users specify where blocking applies
    - ○ Features:
        - ■ Whitelist: Sites where blocking is disabled (e.g. news sites or forums)
        - ■ Blacklist: Sites where blocking is always active (e.g. YouTube, Twitter)
        - ■ Per-site customization (e.g. "Block only images on Reddit")

3. Optional (but valuable)

- ● Features:
    - ○ User-friendly onboarding process
        - ■ Provide an interactive setup wizard when the extension is first installed
        - ■ Prepopulate popular spoiler lists (e.g. trending TV shows/Movies)
    - ○ Statistics and history
        - ■ Show users how many spoilers have been blocked
        - ■ Option to review or undo specific blocking actions (e.g. adding a "Recently Blocked" list)

- - Cross-device syncing
    - Sync user preferences across multiple devices through a cloud service or browser account
  - Community integration
    - Allow users to share or import spoiler lists from other users (e.g. trending TV shows/Movies)
    - Integrate with community-voted keyword list to stay up to date on popular topics
  - Custom alert sounds/visuals
    - Provide themed alert options (e.g. "Caution!" sound or animation")

4. Technical Considerations

Browser Compatibility

- Ensure support for major browsers like Chrome, Firefox, Edge, and Safari
- Use WebExtensions API for cross-browser compatibility

Performance Optimization

- Ensure the extension doesn't slow down the browsing experience
- Use asynchronous operations and minimize the impact of real-time scanning

Privacy Assurance

- Clearly community how data is handled (no tracking, minimal local processing)
- Avoid sending user data (e.g. browser history or detected keywords) to external servers unless absolutely necessary