

- **User Story #1.1:** As a YouTube viewer I want to make a comment on restricted videos and see it
  - **Acceptance Criteria:**
    - User should be able to see comments on restricted videos
    - Posting a comment should interact with the database
    - Able to have HTML overlay
    - Setup database and get basic schema
    - Have a web server with initial endpoints
    - Have server interact with database
- **User Story #1.2:** As a user, I would like to use the app as Chrome extension
  - **Acceptance Criteria:**
    - User should be able to use Chrome extension and have the functionality of a web app
    - Have the frontend app as Chrome extension
- **User Story #2.1:** As a Youtuber I want my viewers to be able to give me feedback even in the case my comments are disabled
  - **Acceptance Criteria:**
    - A comment written by a user should be stored for others to see
    - Link the frontend overlay to the backend to store comments
- **User Story #2.2:** As a commenter I want to be able to be identified by my username
  - **Acceptance Criteria:**
    - Users should be able to sign in with their Google Accounts and see their profile pictures and usernames
    - Use Google OAuth to access and use user's email profile picture and username
    - Map unique user-ids to display names
    - Implement Google Sign-In to grab usernames from Google accounts
- **User Story #2.3:** As a commenter I want to be able to reply to others
  - **Acceptance Criteria:**
    - Users should be able to interact with other users by replying to them
    - Replies should appear as subcomments to the original comment
    - The reply endpoint and schema are setup in the backend
- **User story #3.1:** As a YouTube user I want to be able to comment and see other people's comments in real time [5]
  - **Acceptance Criteria:**
    - Users can see posted comments without refreshing the page
- **User story #4.1:** As a YouTube Comment Reader I want to be able to like/dislike comments [2]

- **Acceptance Criteria:**
  - Users should be able to interact with other users by liking/disliking a comment
  - Create backend request to get likes/dislikes
  - Setup web server endpoints to pass likes/dislikes
  - Pass like/dislike updates from frontend to backend
- **User story #4.2:** As a YouTube user I want to be able to sort comments by date or popularity [1]
  - **Acceptance Criteria:**
    - Users should be able to view comments by most recent, or by most popular
    - Have a button that change the sorting by date or by popularity

#### **Known Bugs:**

- Two users are unable to use the Chrome extension at the same time as Google Cloud only allows one extension ID to be used at a time.
  - In order to fix this, we would need to have the extension validated and approved by Google.
- Comment state does not change when changing videos on the same tab causing comments from other videos to be displayed on an unrelated video
  - A quick refresh will display the correct contents
- All sample commenters have the same profile picture

#### **Product Backlog:**

- Profanity / Hate Speech Filter
- Don't leak implementation details in error messages