**Test strategy document**

**Test scope:**

Scope is to perform the front end web application testing for the BBlog site. System testing is the first level testing to be done.

**Functionality planned to keep in scope:** CRUD articles

*Reason*: As a blogging website, the main business functionality would be for people who write the blogs and users who read the articles should be able to use them seamlessly. Hence the priority is high for this and also this seemed like the be 1st functionality to develop among other general functionalities mentioned. Hence choose this for assignment.

**Assumptions and precondition:**

User has a valid account for the web app and signed in before every test.

Article should have atleast title and body.

Multimedia content should be allowed.

**Some Suggestions**:

Searching an article by name or tag using a text field would be useful rather than going page to page or scrolling down the long list of tags to find.

Sorting by article name and author name logic can be added.

Notifications for comments and favorites can be added.

Inside the article page, the follow and favorite buttons are duplicated in a single page which seems unnecessary.

**Test scenarios and specification:**

Decisions made to choose these test cases for automation and manual:

*Create Article:*

For this I considered positive & negative validations of checking not only the publish article functionality but also what type of content it allows, length of content allowed. Since as a blogging website the fields should accept multimedia content and stop the invalid content from getting published so the site is not corrupted and loaded with unnecessary data.

*Read article:*

For this functionality, it is verified if an article can be viewed from various ways present in the app, like from ‘Your feed’, ‘Global feed’ ‘Using Tags’, or from inside a user profile - ‘My posts’ & ‘Favorited posts’.

I covered 3 flows of this in automation test cases and in manual checked all.

*Edit article:*

Verified if the edit function is working and it’s only allowed to articles owned by the signed in user.

*Delete article:*

Verified if the deleted article is removed from published places.