**USER STORY**

**Requirement:**

Implementation of Create Post Page

|  |  |  |
| --- | --- | --- |
| Epic | User Story | Acceptance Criteria |
| As an **Admin, I Want** to **create a post** by entering information into the form and **store the data** in Local Storage. | 1. As an Admin, I want multiple input fields so that I can add information accordingly for each post. | * The form should take Title, Description, Category, Type etc. as inputs from users. * There should be an input field to take http URL links that should be used as posts background image. * There should be a checkbox to which when checked should display the post in the featured tab in the Landing page. |
| 1. As an Admin, I want to use a feature-rich text editor to format and style content easily | * Rich Text Editor should be used to take formatted data from user which needs to be displayed on detail page of that post. * The Rich Text Editor should be able to align text, add images, add coloring, implement formatted code etc. which should stay as it is when displayed. * The Rich Text Editor should have Menu bar and Toolbar for different kinds of formatting and styling. |
| 1. As an Admin, I want each post to be unique and provide submission time and date for that post | * Each post created should have its unique Post ID. * The submission time and date should be the time when the post was created. |
| 1. As an Admin, I want to store the inputted data into the Local Storage so that it acts as a Database | * The data retrieved from the Create Post form should be stored in local stored with a name. * The data should be stored as JSON based array objects. * According to the post types segregate the data into three sub-arrays and the data should be stored in the Local Storage respectively. |
| 1. As an Admin, I want to be informed of any errors or issues during the post creation process. | * No input field should be left empty when form is submitted and its show warning whenever any input field is left empty except Featured. * Category should always be selected from any of the three types. |
| As an **Admin, I Want** to **display the data** that is stored in Local Storage and display in their described section **based on the Type of Post.** | 1. As an Admin, I want to display the post in a particular section according to its Post Type | * If the Post type is ‘Recent Post’ then it should be displayed in the Recent Post Section adapting the styling of that component section. * Similarly, for ‘Sponsored News’ and ‘Top Highlights’ posts should be separated according to their post type and displayed in their described section. * A post should not be rendered in other section. It should be unique. * Latest four or six posts should be displayed on the Landing Page even if there are more posts in that category. * If there is no data in the local storage only the section headings should be displayed. |
| 1. As an Admin, I want the ability to mark a post as featured to increase visibility among other posts on the platform. | * Only the latest four post should be displayed. * The most recent featured post be displayed in the big box and if a new post is added the post should be post to the second box. |

**USER STORY**

**Requirement:**

Implement Header and Footer Section

|  |  |  |
| --- | --- | --- |
| Epic | User Story | Acceptance Criteria |
| As a **User, I Want** the application to display a visible and consistent header  and footer section on all the pages | 1. As a user, I want to click on menu button so that I can see the hamburger menu details. 2. As a user, I want to see the footer section of the application to access important information. 3. As a user, I want to subscribe to post directly from the footer section by providing my email address so that I can get the regular updates. | * On the application, the menu button should be represented by an icon such as three horizontal lines. * When user click on the menu button, a responsive hamburger menu should appear on the screen. * The hamburger menu should overlay the existing content. * The menu items within the hamburger menu contains navbar content and subscribe button. * The menu should be static to the page. * The hamburger menu should have a smooth transition when opening and closing enhancing the user experience. * The application should consistently display a footer section at the bottom of each page. * The footer should contain relevant information and subscribe button or copyright information. * The footer section should include a distinct subscription form. * The subscription form should include a labeled input field for entering the user’s email address. * The form should have a “Subscribe” button. |

Functional Validation of Subscribe

1. Verify that system does not accept the empty submission and prompts the user to fill in the required fields.
2. Verify that the email field displays an error message when not in the correct email format.

**USER STORY**

**Requirement:**

Implement Detail Page

|  |  |  |
| --- | --- | --- |
| Epic | User Story | Acceptance Criteria |
| 1. As an **Admin**, **I Want** to open the detail page **so that** I can see the full description and additional details on each post. 2. As an **Admin**, **I Want** to add the comment section **so that** I can add the comments on each post. | 1. As an Admin, I want to click on the post so that I can navigate to the detailed page of that post. 2. As an Admin, I want to view static data on the post detail page when I directly navigate to it without clicking on a specific post. 3. As an Admin, I want to be able to add comments to a post on its detail page. 4. As an Admin, I want to see existing comments so that they do not disappear on page refreshes. | * When admin clicks on the post in the landing page, the application should navigate to the detail page of the specific post. * The detail page should display comprehensive information of the post what admin had included in creating the post. * The navigation back to the landing page should be user-friendly. * When no post exists on the landing page and admin directly navigate to detail page using URL. The page should display static data. * The page will display header, footer, and relevant message that no post exists. * Comment form will display zero comments and all input fields and button will be disabled. * On the post detail page, there should be a section for adding comments. * The admin should be able to enter name, email address and email address in the text input field. * There should be “Post comment” button to post the comment. * After submitting the comment, the comments should be displayed above the comment form. * Each comment should show the admin’s profile, username, timestamp, and the comment text. * When admin adds the comment, they should store it in local storage. * Number of comments in a post should display. * When admin navigate to post detail page, existing comments associated with post should retrieve and display. * The comment section should remain visible when the page is refreshed. * The comments should dynamically update when any new comment is added. |

Functional Validation of Comment Section

1. Verify that all the input fields are mandatory.
2. Verify that tab functionality is working properly or not.
3. Verify that system does not accept the empty submission and prompts the user to fill in the required fields.
4. Verify each input field, on submitting empty field.
5. Verify that the system displays an error message for each empty field, indicating that it is mandatory.
6. Verify that the required input field have (asterisk\*) next to the label, to indicate their mandatory status.
7. Verify that the email field displays an error message when not in the correct email format.

**USER STORY**

**Requirement:**

Implement Log In Page

|  |  |  |
| --- | --- | --- |
| Epic | User Story | Acceptance Criteria |
| As an **Admin**, I want to securely log in to ensure access to confidential information and system controls. | 1. I want to design a login form featuring two input fields, one checkbox and single button. | * The form design should be responsive, ensuring usability across different screen sizes and devices. * Ensure that the checkbox state and entered information in the input fields persist if there are errors upon form submission, maintaining user-entered data. * Implement visual cues to indicate the active state of input fields and the button (e.g., highlight or change in appearance when focused or hovered). |
| 1. I want to log in with my registered username and password, so that I can access my account and its associated functionalities. | * Successfully log in only if the entered credentials match the registered admin credentials. * Ensure that access is granted solely for registered admin users with their specific credentials. |
| 1. I want the system to verify the email format entered in the username field, so that I can ensure the correct email format is used for authentication. | * Ensure the system checks the email format before authentication. |
| 1. I want to receive clear feedback when entering incorrect credentials, so that I can understand why the login failed and take appropriate action. | * Promptly inform the user of the reason for login failure due to incorrect credentials. * Ensure the error message is concise, clear, and immediately understandable to the user. * Maintain the login form's state, preserving entered data for correction without requiring re-entry. |
| 1. I want to stay logged in across sessions by checking “Remember me” checkbox. | * Return to the application later and remain logged in without re-entering credentials. * Directly access the personalized dashboard or homepage upon returning without going through the login process. |
| 1. Provide the ability to store my login information, allowing me to conveniently access saved details the next time I visit the login page.   Top of Form | * Encounter a pre-filled login form with previously saved details upon revisiting the login page. * Successfully log in using the saved details without re-entering the credentials manually. |
|  | 1. I want to experience distinct functionalities and navigation based on my login status. | * I want the application to set default admin credentials in the local storage. * I expect the application to update the admin login status and store this information in the local storage. * Before Admin logged in, If I try to access admin-specific pages directly by typing the URL, then I expect the application to redirect me to the login page for authentication. |

**Login Page Functional Validations:**

* Verify that the form components (input fields, checkbox, button) maintain usability and alignment across different viewports.
* Confirm the active state changes (e.g., highlighting) of input fields and the button upon interaction.
* Ensure the error message explains the reason for login failure and retains entered data for correction without requiring re-entry.
* Input various email formats into the username field and confirm that the system checks and verifies the format before authentication.
* Confirm successful login using saved details without manual re-entry.
* Verify that the application updates admin login status and stores this information locally.

**USER STORY**

**Requirement:**

“Load more post” button functionality.

|  |  |  |
| --- | --- | --- |
| Epic | User Story | Acceptance Criteria |
| Enhanced content exploration with “**load more post**” button. | * 1. I want the 'Load more post' button to allow me to view additional images related to today's top highlights without navigating away from the current page. | * Upon loading the page, a subset of images related to today's top highlights should be visible. * The 'Load more post' button should appear if the total number of available images exceeds the initially displayed set. * Clicking the 'Load more posts' button should trigger the display of the next set of images related to today's top highlights without a page refresh. * The new set of images should be appended below the existing displayed images. * The 'Load more posts' button should disappear, when all images have been displayed. |

**USER STORY**

**Requirement:**

Implement Post Listing page.

|  |  |  |
| --- | --- | --- |
| Epic | User Story | Acceptance Criteria |
| 1. As an **Admin**, **I Want** a table with all the posts, **so that** all the posts and its category can be viewed, created, updated and deleted. 2. As an **Admin**, **I Want** to implement search, sort and pagination functionality **so that** all the posts can be easily accessed. | 1. As an Admin, I want to create a table so that all the posts can be viewed. 2. As an Admin, I want to have a create post button, so that it navigates to the create post page. 3. As an admin, I want a create post page , so that I can create new posts. 4. As an admin, I want to a delete button, so that I can remove the posts that are not necessary. 5. As an admin, I want to an edit button, so that I can edit the posts that are already posted. 6. As an admin, I want an edit post page, so that I can edit the posts and upload again. 7. As an admin, I want a search functionality, so that I can easily search for the posts I want. 8. As an admin, I want a Sorting functionality, so that I can look for the latest posts, oldest posts or alphabetically. 9. As an admin, I want a pagination functionality, so that I can see only 10 posts per page. | * All the posts should be displayed in the table whatever exists in local storage. * Once the admin logs in, this page should be displayed first. * The table should consist of the following columns: Post Name, Submission Day, Category and Actions * If admin clicks on the create button, it should navigate to the create post page where all the details of the post is entered. * The create button should navigate to the create post page with all the required fields. * On clicking the create post button on create post page, the new posts should be displayed in the Post-Listing Page and Landing Page. * In the actions, there should be a delete button which will delete that post from the table. * The same should reflect in the landing page and local storage. * The edit button should navigate to the edit post page. * The edit post and create post page should be the same code, based on the button click it should show create post or edit post. * The edit post should have prefilled data in all the inputs, that was previously stored (fetching from the local storage) * After update button is clicked, the submission date should not change only the data that was changed should display in the detail page, landing page and the Post Listing Table. * The input box should take input and then search for the string that matches that post name. * The search should also include the category. * The dropdown box should have 3 options which will sort them according to Latest Posts, Oldest Posts or by Name. * The Name option will sort the posts alphabetically. * The latest posts will sort the list as newest posts first and the Oldest Posts will sort the list as oldest posts first. * By default, the posts should be sorted as newest first. * The pagination functionality should display only 10 posts per page. * On the right end, there should be a next and previous button to see the next 10 or previous 10 posts respectively. * In between the next and previous button, it should also display buttons with total number of pages. * If the first page is displayed, the previous button should be disabled. If the last page is displayed the next page should be disabled. * On the left end, it should show how many out of how many total pages are being displayed. * If there are no posts, then it should display “Showing 0 out of 0 pages”. |