

One of our prominent clients is willing to start a new venture. They are trying to get into the Publishing platform. As a result they want us to build the platform.

Development phase is divided into two phases. First phase will contain the Minimal Viable Product (MVP). Once the client is content with the MVP, the next phase will contain more advanced features.

As an Engineer you have the freedom to choose tools and technology required to implement the required features.

Requirements (**MVP**):

Client requires a browser based app, from there editors and journalists can create textual contents.

Frontend:

- a. It needs to be a single page application.
- b. Users would require authentication to get into the application. So there needs to be a login screen.
- c. Once the user logs in, he/she will see the Dashboard page, containing a list of already created content. There will be a logout link on the right side corner. Clicking Logout will clear the session for the user and the app will navigate to the login page.
- d. There should be a link/button in the initial screen labeled "Create".
- e. Once the "Create" button/link is clicked, navigate to a page to create content.
- f. The Create content form should contain two fixed fields, Title and Body. Title will be a plain text field and Body will be a rich text field.
- g. Body field will need a toolbar, containing the option for making text bold and italic.
- h. There will be a "Save" button, below the form, once clicked content needs to be saved and navigate to the initial page and show the newly created content in the Dashboard (mentioned in step c)
- i. List only shows the content title, creation date and an edit icon.
- j. When the edit icon is clicked, navigate to the edit content form, where Title and Body will be populated with the content's title and body. The Editor/Journalist can make some changes in content and "Save" the content. After hitting Save, navigate to the Dashboard.

For implementing the front-end you can utilize Typescript/ Javascript. It would be great if you can avoid using libraries and framework as much as possible. Just add strictly necessary things.

Backend:

- a. Build a backend with RESTFul API, to accommodate creating and editing content.
- b. Implement Authentication
- c. There will be two roles, Journalist and Readers. Journalist will be the only role who will be able to login in the frontend app. Journalist will create and edit content.

- d. Reader on the other hand will only be able read the content. No create and edit capability.

Choose any language between Java, Python or Javascript. For choosing DB, both RDBMS and ObjectDB options are open.

Website:

- a. Build a minimum website that can show the content created through the editor.
- b. There will be no requirement for authentication.
- c. Website will have basically two pages, index page will contain a list of contents, list item should contain title and creation date and first 100 characters from body field. And when a visitor clicks on the title of the content, navigate to the details page of the content. Details page will show everything from title to body field, creation date and journalist who wrote that piece.

Advanced Features:

Frontend:

- a. Journalist/Editor will be able to embed pictures in the richtext.
- b. Journalist/Editor will be able embed youtube videos and Tweets in the richtext.

Backend:

- a. MVP defines only one type of content that has two fixed fields. For advanced features, it should be possible to define a content model through an xml/json/yaml. System administrators should be able to define a xml/json/yaml file denoting how many fields the content will contain and the type of the field, basic or rich text. According to the content model, the frontend will render the form.
- b. As the API designated for the website can be utilized by some other third party application, like a mobile app or other content provider, the system needs to be able to impose rate limiting on API calls. System administrator will be setting up the GET request rate per minute per client.

Please try to finish up the MVP first before moving into the more advanced features.