Table of Contents

Functional & Non-functional requirements

Functional requirements

User Management:

- The system will allow users to register with email, username, password, and role.
- The system will allow users to reset password with reset tokens valid for 5 minutes.
- Superusers can assign **journalists/editors to publishers** via the admin site or the app interface.

Roles & Permissions

- The system must distinguish between **Reader, Journalist, and Editor** roles.
- Journalists must be able to create articles and newsletters for a publisher or independent content.
- Editors must be able to approve or reject articles and newsletters as well as read, update and delete.
- Readers must be able to subscribe to:
  - Publishers
  - Journalists
- Readers must be able to view articles and newsletters that are published, but must be subscribed. They just won't receive emails for the published content if they are not subscribed to the publishers.

Article Management

- Journalists must be able to create & publish independent articles.
- Journalists must be able to submit articles to a publisher for approval.
- Editors must be able to approve articles before publication.
- Readers must be able to view all approved articles provided they are subscribed.

Newsletter Management

- Journalists must be able to create newsletters and add articles to that newsletter, but pertain to one publisher for a newsletter creation.
- Editors must be able to approve newsletters for publication.
- Readers must be able to view published newsletters based on their subscription preferences.

Subscriptions

- Readers must be able to subscribe or unsubscribe from:
  - Journalists
  - Publishers
- Readers must receive approved articles only from publishers they are subscribed to.

Content Delivery

- Readers must be able to view:
  - Articles
  - Newsletters
- Content feed (combined feed based on subs)

Signals/Views

- The system will either use signals or views to make posts to twitter.
- The system will perform post migrations signals, to create groups needed, as well as assign the correct permissions to the groups.

Authorisation Decorators

- The system will have custom decorators for the authorization depending on the logged in users' role and group.

API Functionality

- Readers will have to login to the web service.
- Will allow clients to make GET request for articles for publishers and journalists, that the client is subscribed to in JSON format.

Non-functional requirements

Performance

- The system must respond to API requests within 2-4 seconds.

Security

- The system must securely store passwords using hashing (Django default).
- Only authenticated / authorised users may access protected endpoints.
- Users can only perform actions allowed by their role & authorisation.

Reliability

- The system must not lose data after crashes.
- All published content must always be retrievable.

Scalability

- The system must support multiple publishers, journalists, and readers.
- The design must allow future features without breaking existing structure.

Usability (UX)

- The application must be easy for non-technical users to navigate.
- The UI must be responsive on desktop and mobile.

Maintainability

- Code must follow Django best practices (apps, models, serializers, views).
- API will be documented with DRF docs.

UI/UX Planning

UI Screens Needed

Authentication

- Login Page
- Register Page
- Forgot Password Page

Reader UI

- Home Feed (articles + newsletters)
- Article Details Page
- Newsletter Details Page
- Subscriptions Page

Journalist UI

- Dashboard: My Articles
- Create Article Form
- Edit Article Page
- Create/Edit Newsletter Form
- My Newsletters Page

Editor UI

- Articles dashboard.
- Review + Approve Article Page
- Newsletters dashboard.
- Review + Approve Newsletter Page

Admin / Publisher Management

- Publishers List Page
- Assign Editors/Journalists Page

UX Considerations

- Consistent navigation across roles
- Minimal clicks to perform main tasks

- Clear visual distinction of:
  - published vs draft articles
  - approved vs pending content

- Success + error messages after major actions