MAGAZINE CHALLENGE

trivago

ASAD MANZOOR

Application Architecture

Single Page Application:

The Application is divided into following components.
- Listing Component
- Single List Item Component
- Single Detail Component

Home Page Trending Articles:

- Home page has an overview of the latest articles from the /latest_posts.json.

- For each article the title, the thumbnail (image), the excerpt of the article are shown.
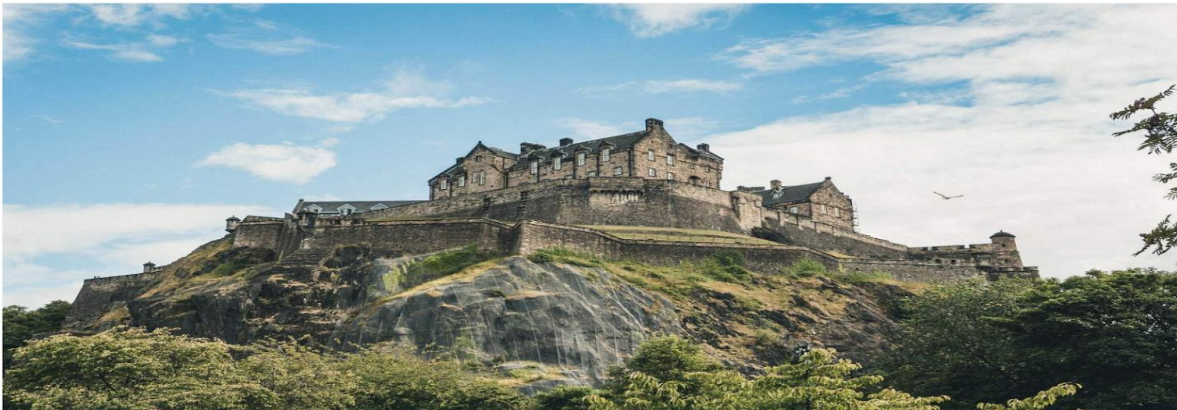
Article Detail Page:

- An article detail page present the entire article fetched from the uri from the /latest_posts.json.

- The article detail page contain the thumbnail (image) on top of the page, the title of the article, and the "intro_text" from the content section 0.

- **Rating** functionality is implemented using django rest framework. API endpoints are exposed and in **magazine** react app these endpoints are used for submitting ratings of articles.

Improve Search Engine Crawling:

Solution:

- Change meta robots : **noindex**, **nofollow** to **Index**, **follow**.
- By this change now you are allowing search engines to crawl the site

Dockerize Application Deployment:

- Created a Django app that runs inside a Docker container.
- Create a React app with the all-too-literally-named Create React App that runs inside a Docker container.
- Implement these dockerized apps as services in Docker Compose.
- Connect the magazine service to a basic backend API from which it can fetch data.

Github Code Link: [Docker Django React Trivago APP](#)