Contents

Written By
Ethan Krimer

App designation

The application will operate as a platform for uploading and reviewing CV files between the members of the Magshimim-Next community.
This will allow us to improve and maximize the potential of every member's CV – by getting a lot of input from various users.

How will it work?

Upon entering the platform, a user will have to register/sign in with a google account (still need to reconsider this).

Once the user is signed in – he will be able to upload as well as view various CVs of the active platform members.

Uploading a CV

A user can choose to upload any link he'd like to share – but a submitted link must withstand the following standards:

- If a .**docx** file is being shared – a user should share it via **Google Drive** with the Comments option toggled for anyone with the link.
- If a .**pdf** file is being shared – a user should share it via **Drop Box** with the Comments option toggled for anyone with the link.

- Once a user submits a link, the latest link is bound to his registered account.
    - This means we can upload a new version at any given time.
- Upon submission – each CV should also be linked to a job category (read more about categories in the data schemes section).

Viewing a CV

A user can view any CVs that he'd like – the CVs will be shown in a feed-like page.

The feed will have filtering options but in default settings – will show the most recently

Uploaded CVs.

There will also be an option to filter by category (read more about categories in the data

schemes section).

Once a user has clicked on a CV, he will be sent to the provided CV link – where he can leave a detailed review using the built in comment tools in Google Drive / Drop Box.

We will also provide a general comment section under each post.

Written By
Ethan Krimer

Technologies

Due to the short time schedule on this project and the project being a voluntary work – we will not be creating a server side for this platform but instead use serverless solutions such as Firebase.

- Firebase – Firestore – as the DB of the project.
- AWS – to host the application – preferably with a custom docker image.
- React.next - to code the application.

Schemas & Collections

Categories (Enum)

1) General
2) Medical
3) Insurance
4) Financial
5) Legal
6) Education
7) Full Stack
8) Front End
9) Back End
10) DevOps
11) Cyber Security
12) ...

Users (Firebase generated)

- Unique ID
- Email address
- Name
- Created (long epoch)
- Last login (long epoch, nullable)
- Active (boolean)

CVs

- Unique ID
- Document link
- User ID
- Upload date (long epoch time)
- Category ID (from the Enum)
- Description

Written By
Ethan Krimer

Comments

- Unique ID
- Document ID (Null if nested comment)
- Parent Comment ID (Null if root comment)
- User ID
- Last update (long epoch time)
- Upvotes (list of user IDs)
- Downvotes (list of user IDs)
- Data (string)
- Resolved (bool) – only admins / document owner ID can resolve.
- deleted (bool) – owner / admin / document owner can toggle.

General Workflow UML

Classes UML

Written By
Ethan Krimer