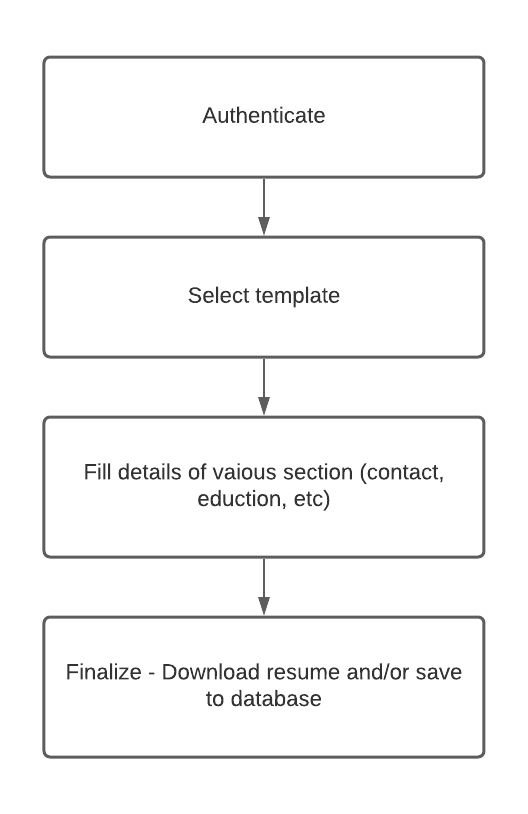
**Features (Resume builder app)**

In this section we will see the features available on the reels app and how they work. The **technical** aspects (tech stack specific details) are discussed in the **next section**. The work flow on this app goes as follows:



1. **Authentication:**

Email and password-based authentication is used. Once a user is logged in (through signup or login), then the authentication details are saved in **cookies**. These cookies are then used for **persisting** the logged in state of the user in case the page is manually refreshed or the web app is opened in a new tab. Once the user logs out of the web app, these cookies are reset. The user auth data is also saved in the redux-store which we will explore in the technical section.

1. **Templates:**

On “*/getting-started”* page, various resume templates are present. These templates define “skin” of the template design. Depending on which template is chosen, the styling of the resume will be defined.

1. **Resume sections:**

After choosing the template, user fills various resume sections like **contact** section and **education** section. The details filled **will not be cleared** if the resume template is changed after the form fields have been filled. The form fields and preview sections are shown side by side.

1. **Finalize:**

At this final stage, the resume could either be downloaded (in **pdf** format) or saved to database.

**Technical details**

This section describes which components of the tech stack were used to implement the above features.

1. **Redux:**

All the data that is required by several components is maintained in the redux store. The store contains resume specific details (like id, skin code), resume sections data (for example, contact section) and data related to firebase. The firebase data in store is handled by functions and Provider given by **react-redux-firestore** and **redux-firestore** packages. The “firebase“ state in store has auth related data which is used for private routing and conditional rendering of components (for example, Signout button in the header section).

The details of resume sections are saved to the store when user moves from one section to another. These details are mapped to the section fields and are used when a resume is saved to the database.

1. **Authentication**

Firebase auth is used for creating (Signup) as well as logging in users. The persistent login feature is achieved using “onAuthStateChanged” property of firebase auth object. It automatically recognizes (using cookies) whether a user was logged in before and did not log out. The auth data is available in the “firebase” state present in the redux store.

1. **Templates:**

Each template is associated with a unique skin-code. These skin-codes are used for styling the resume in resume sections. A skin-code gets added to the class names of each element. Skin-code specific styling of elements is defined in the **scss** files (explore *documentStyles.scss* for the same).

1. **Resume sections:**

The form field and preview section elements are mapped together so that if one changes, the other changes at the same time. The data entered in the form fields are save in the redux store which will be utilized in the ways discussed above (Technical details – point 1).

1. **Finalize:**

For downloading resume in pdf format, **jsPDF** and **html2canvas** packages are used. For saving the data in database, firestore is used which we get from useFirestore hook. Redux store contains the firestore object which has the related functionalities.