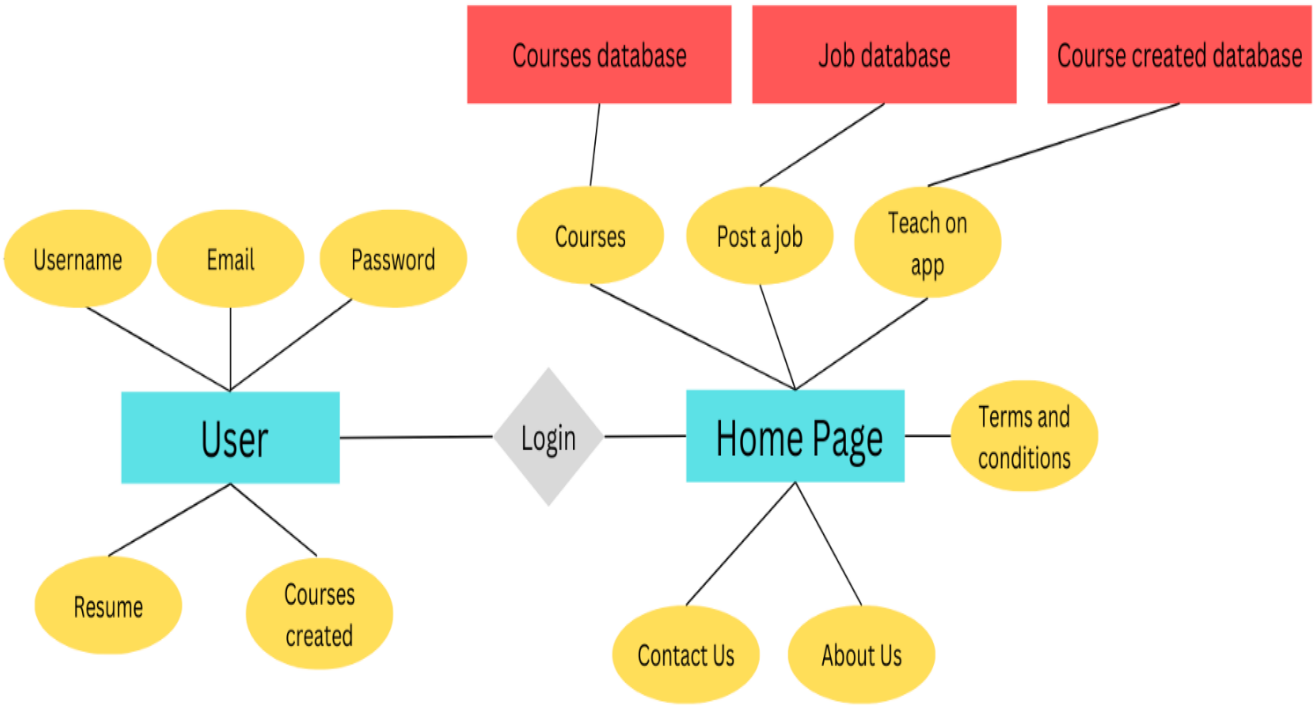
System Architecture



- **Frontend Implementation**:
    - Leveraged the Flutter framework, powered by the Dart programming language, for frontend development. This allowed us to craft a visually appealing and intuitive user interface by leveraging Flutter's extensive library of customizable widgets and layouts.
    - Utilized Dart's reactive programming model to ensure seamless navigation and interaction within the app, prioritizing user experience and accessibility. Dart's declarative syntax facilitated the creation of responsive UI components that adapt to various screen sizes and orientations.
    - Utilized Visual Studio Code (VS Code) as our primary IDE for efficient

coding and development. With its seamless integration with Flutter and Dart plugins, VS Code provided advanced features such as code completion, debugging, and hot reload, enhancing productivity and streamlining the development process.

- **b) Backend Implementation**:

- Integrated Firebase directly into the app using Dart to facilitate courses and job databases for users. Dart's asynchronous programming model ensured efficient handling of network requests, enhancing the app's responsiveness.
- Implemented upload pdf using Dart to accurately identify and curate jobs posted.
  - Leveraging Dart's icons and material libraries to make the UI creative
  - Ensured data security and privacy by employing firebase authentaction. Dart's built-in support for local storage mechanisms allowed us to securely manage and access sensitive user information, maintaining compliance with data privacy regulations.

- ***User Authentication with Firebase Authentication:***

The study focuses on leveraging Firebase Authentication for user authentication in MindMingle. By utilizing Firebase Authentication's robust features such as email/password authentication, social media login, and single sign-on (SSO), MindMingle ensures secure and seamless user authentication processes, enhancing user trust and data security.

- ***Database Creation and Management with Firebase Realtime Database:*** The study delves into the creation and management of three primary databases – users, courses, and jobs – using Firebase Realtime Database. This NoSQL cloud database provides real-time data synchronization across devices, enabling instant updates and consistent user experiences. MindMingle utilizes Firebase Realtime Database to store and retrieve user profiles, course information, job listings, and other essential data, ensuring efficient database management and optimal performance.

- ***Users Database:*** The study explores the structure and functionalities of the users database in MindMingle, which stores user profiles, preferences, authentication credentials, and other relevant information. Firebase Realtime Database facilitates seamless integration of user authentication with user data storage, enabling personalized user experiences and streamlined user management.

- ***Courses Database:*** The study outlines the design and implementation of the courses database in MindMingle, which stores information about the available

courses, including course titles, descriptions, instructors, and enrollment details. Firebase Realtime Database's flexible data model allows for efficient storage and retrieval of course data, enabling users to browse, enroll in, and track their progress in courses seamlessly.

- ***Jobs Database:*** The study discusses the structure and functionality of the jobs database in MindMingle, which stores job listings, employer profiles, job requirements, and application details. Firebase Realtime Database facilitates real-time updates to job listings and seamless integration with user profiles, enabling users to explore job opportunities, submit applications, and track their job search progress effectively.

Fig 1 Home Page UI

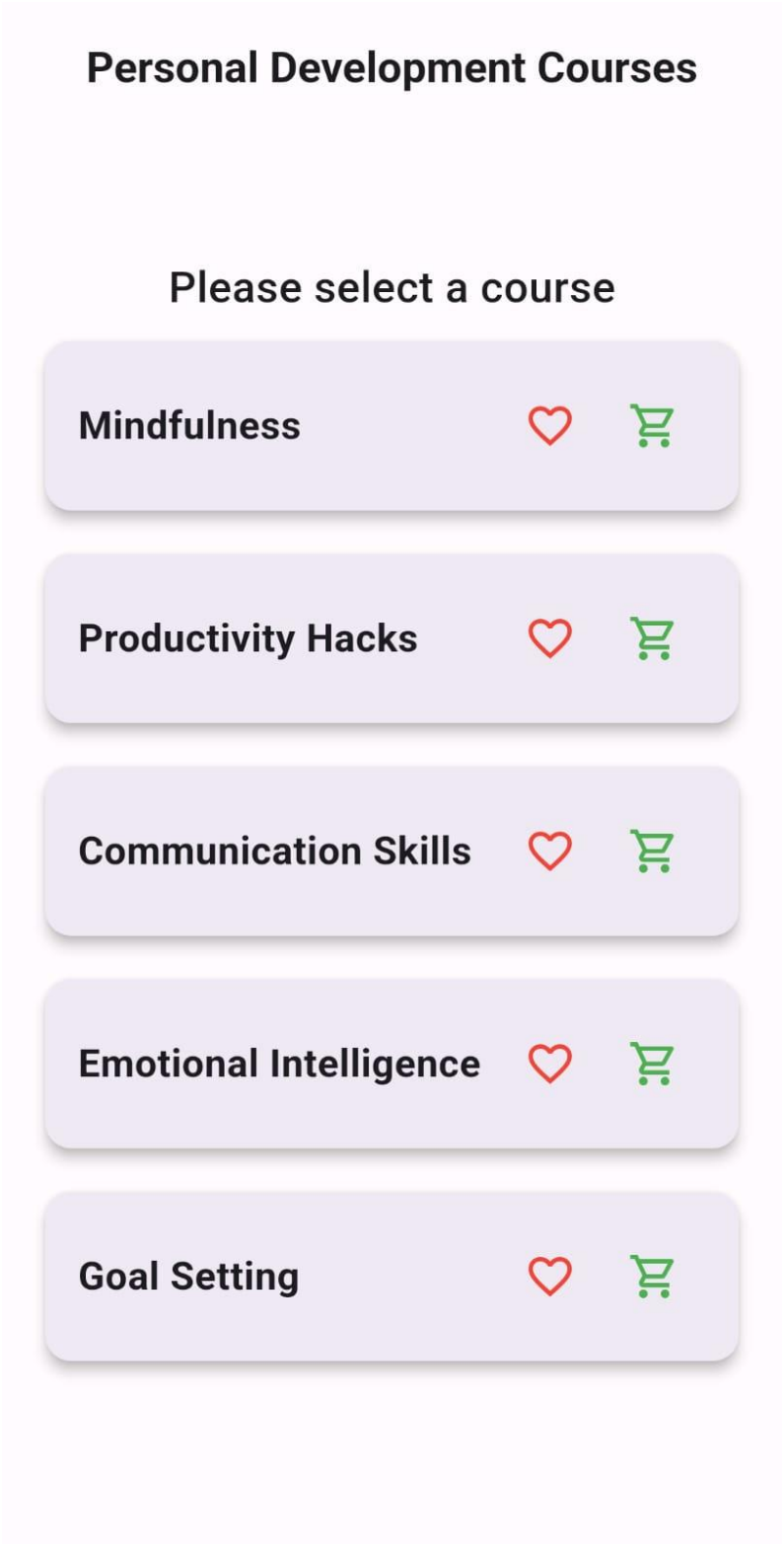


Fig 2 -Courses UI

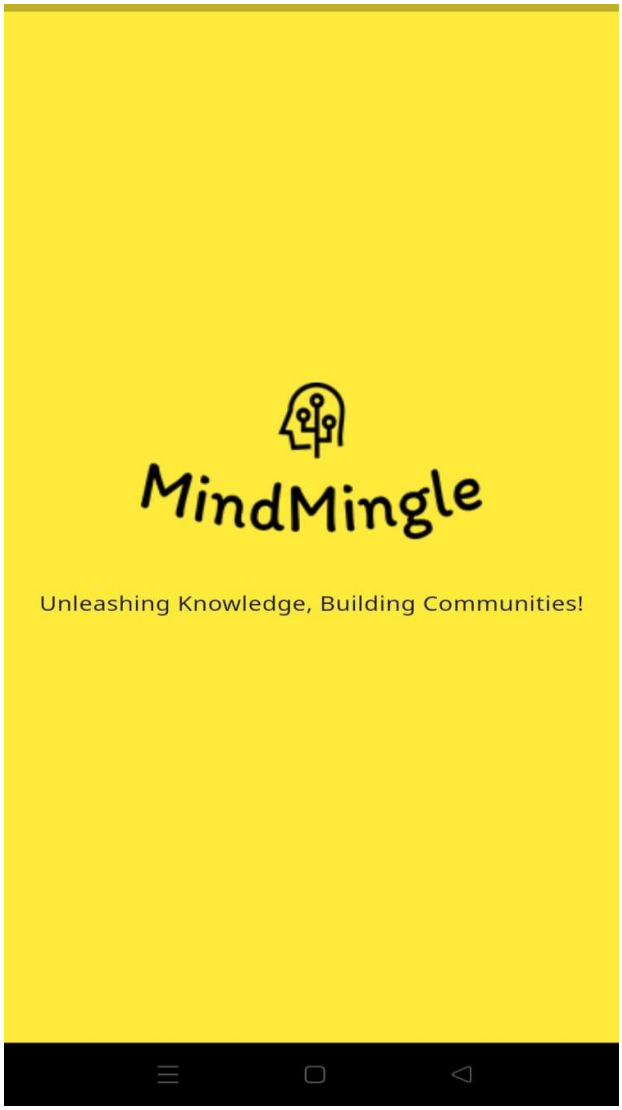


Fig 3-Splash Screen

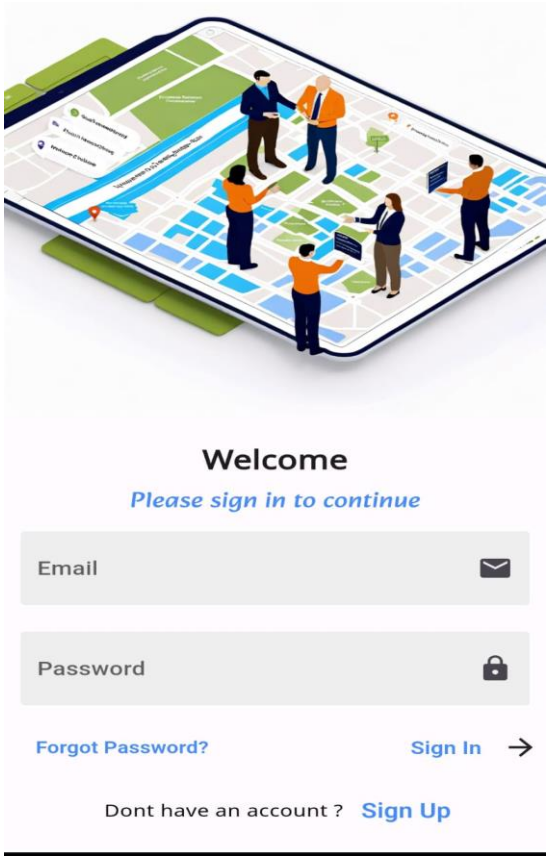


Fig 4- Sign In Page

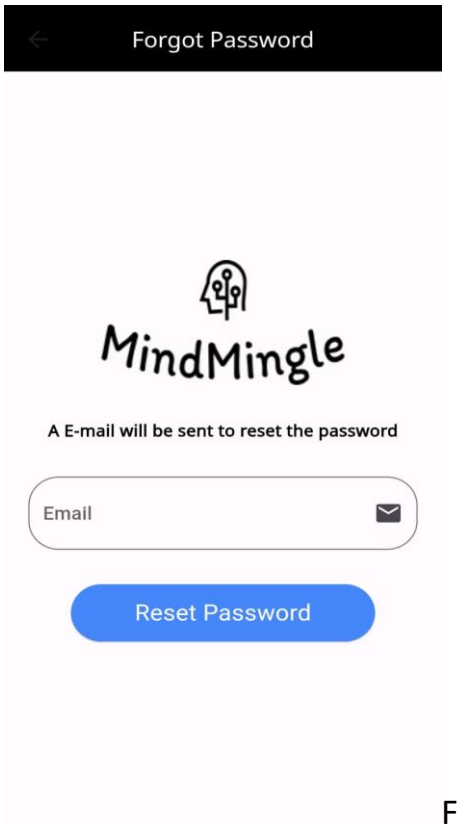


Fig 5- Forgot Password Page

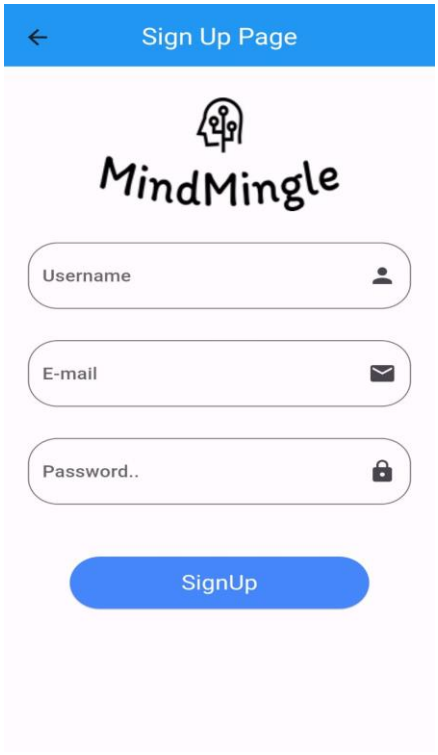


Fig 6-Sign Up Page

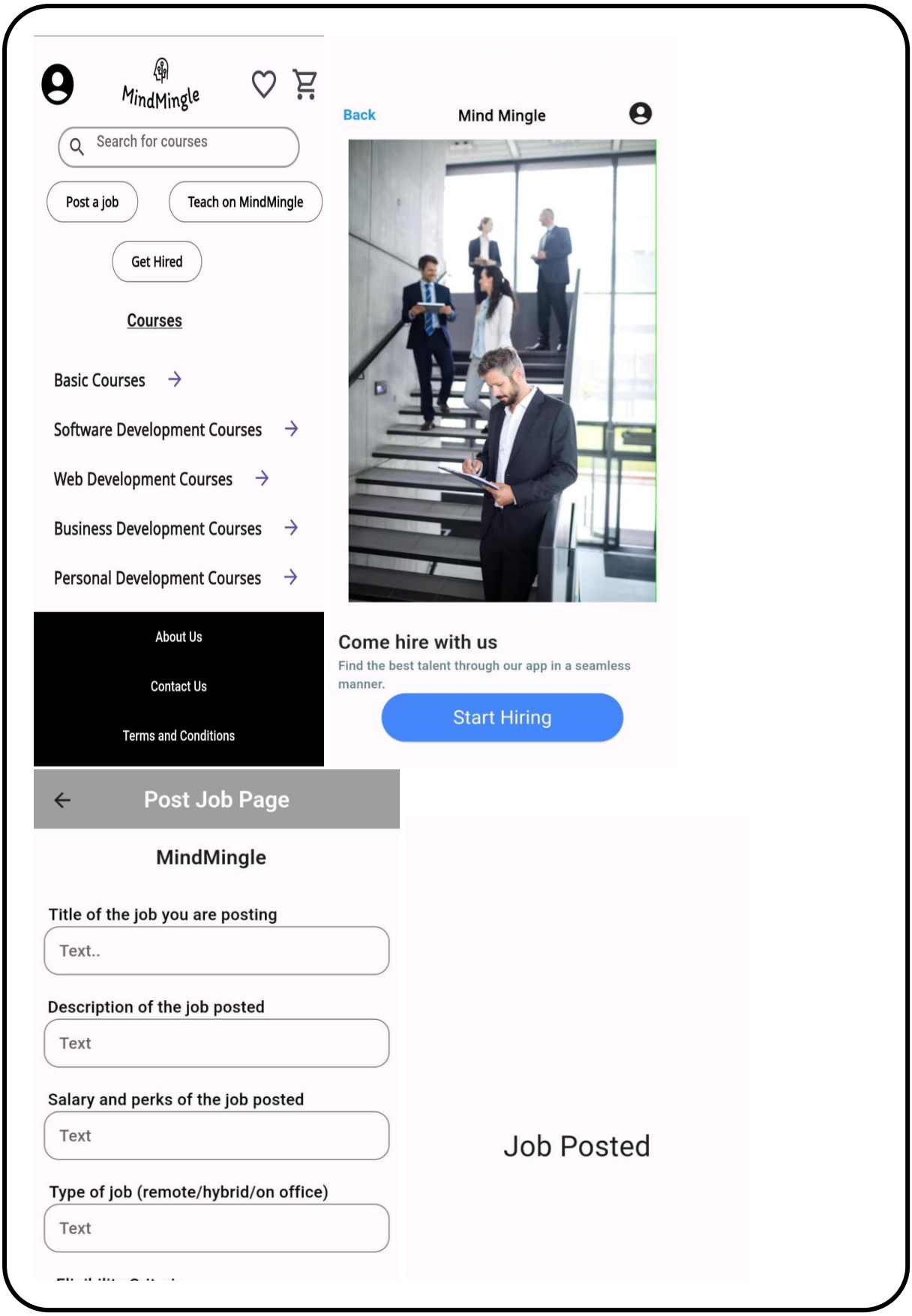


Fig 7- Post a job page

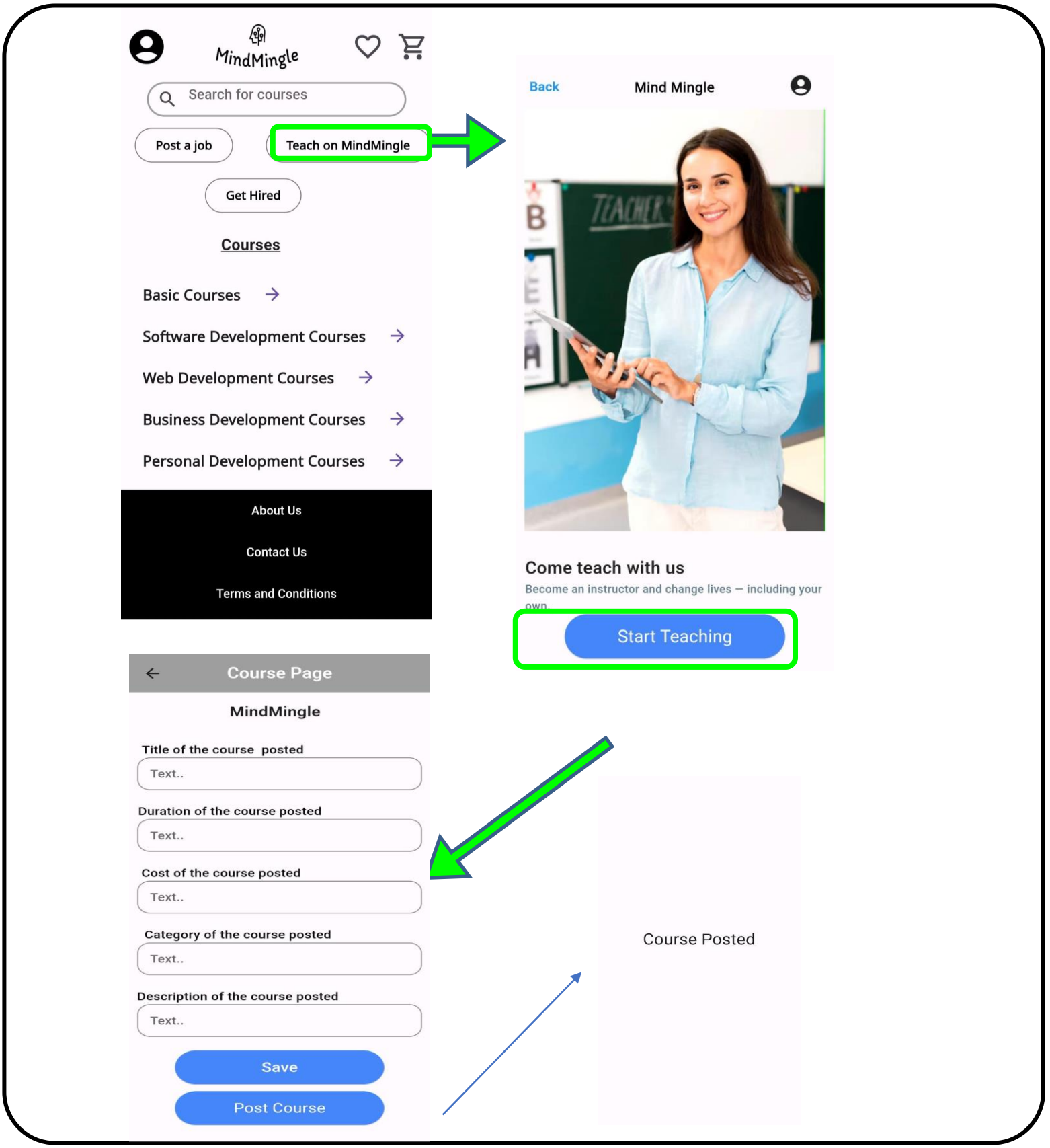


Fig 7-Teach on Mind Mingle

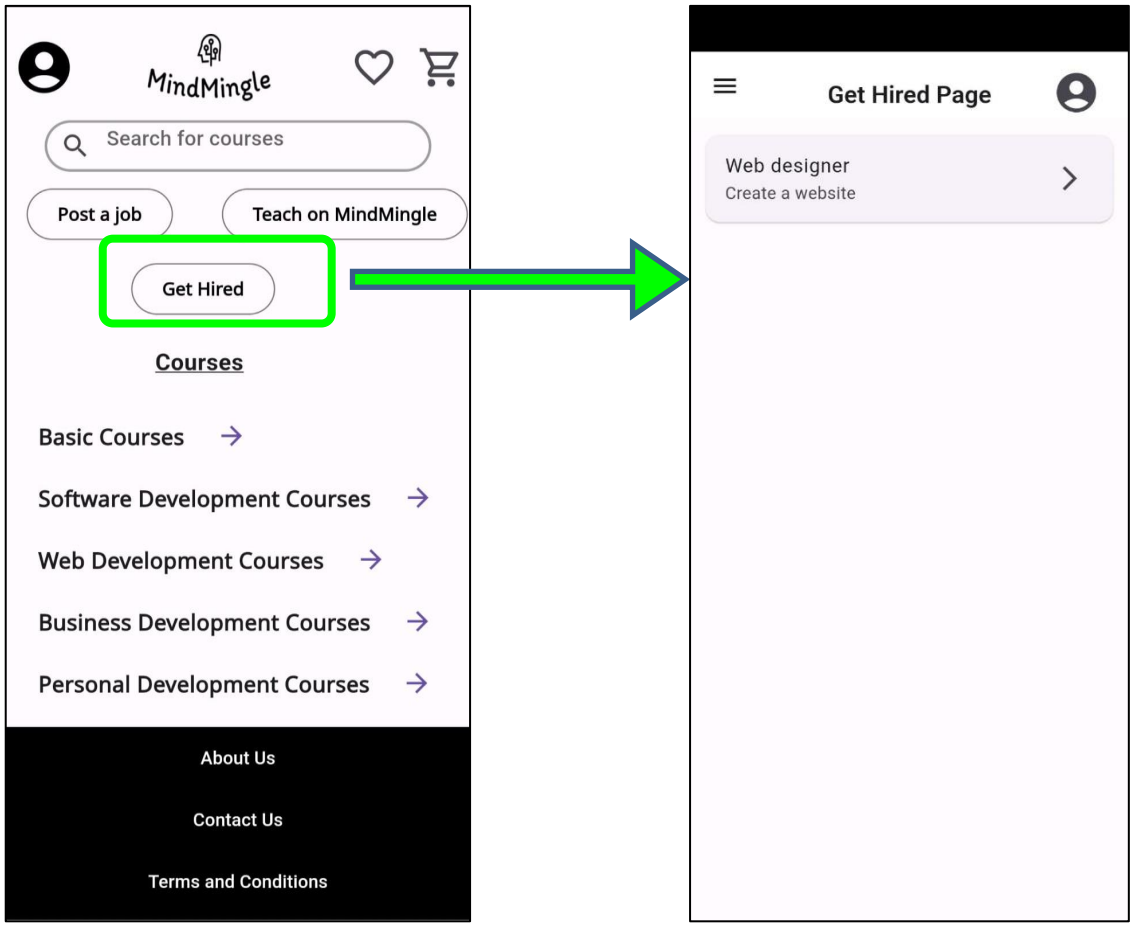


Fig 8-Get Hired Page

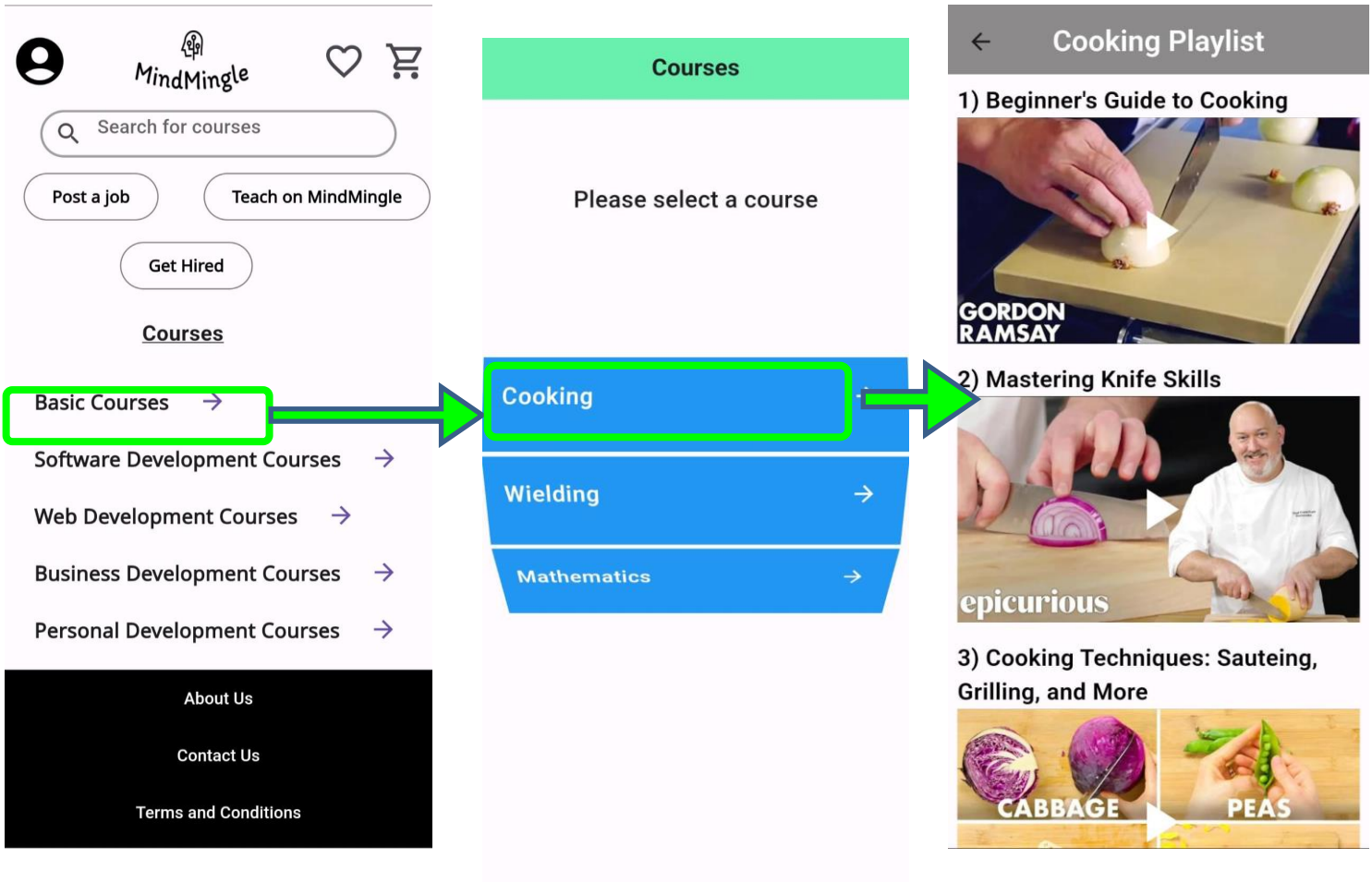


Fig 9-Courses Page

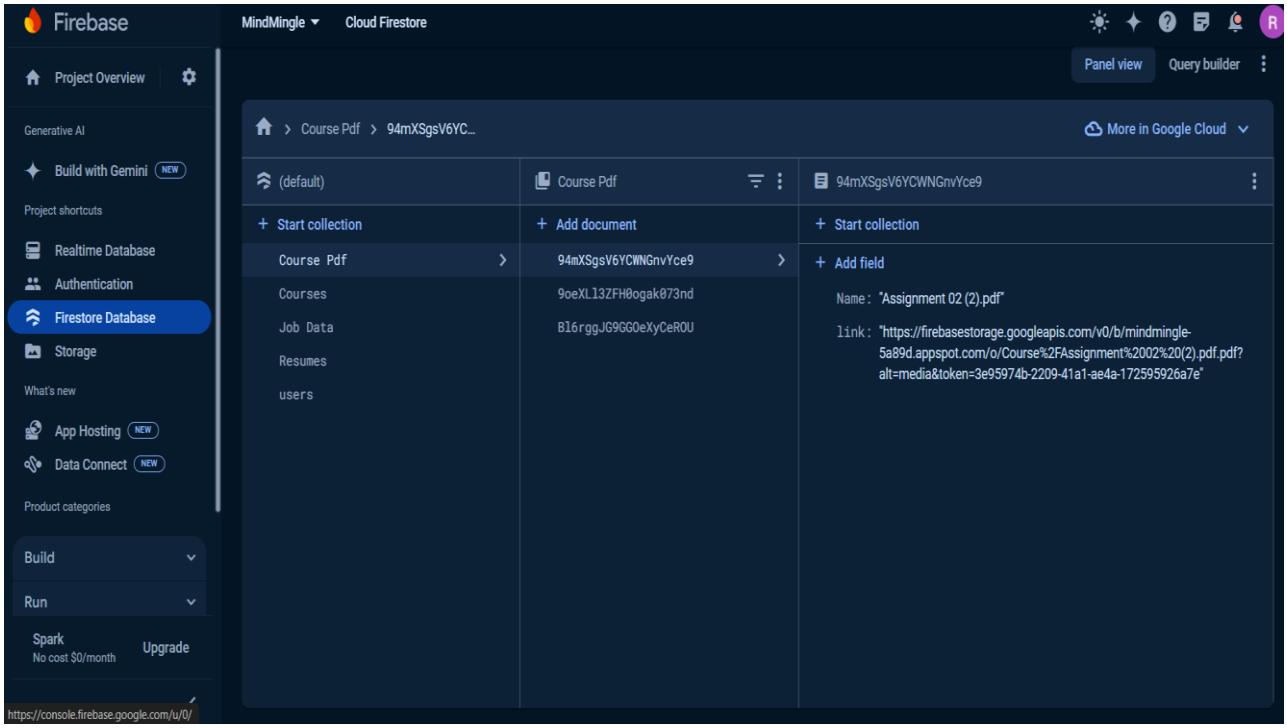


Fig 10 Courses structure pdf upload storage

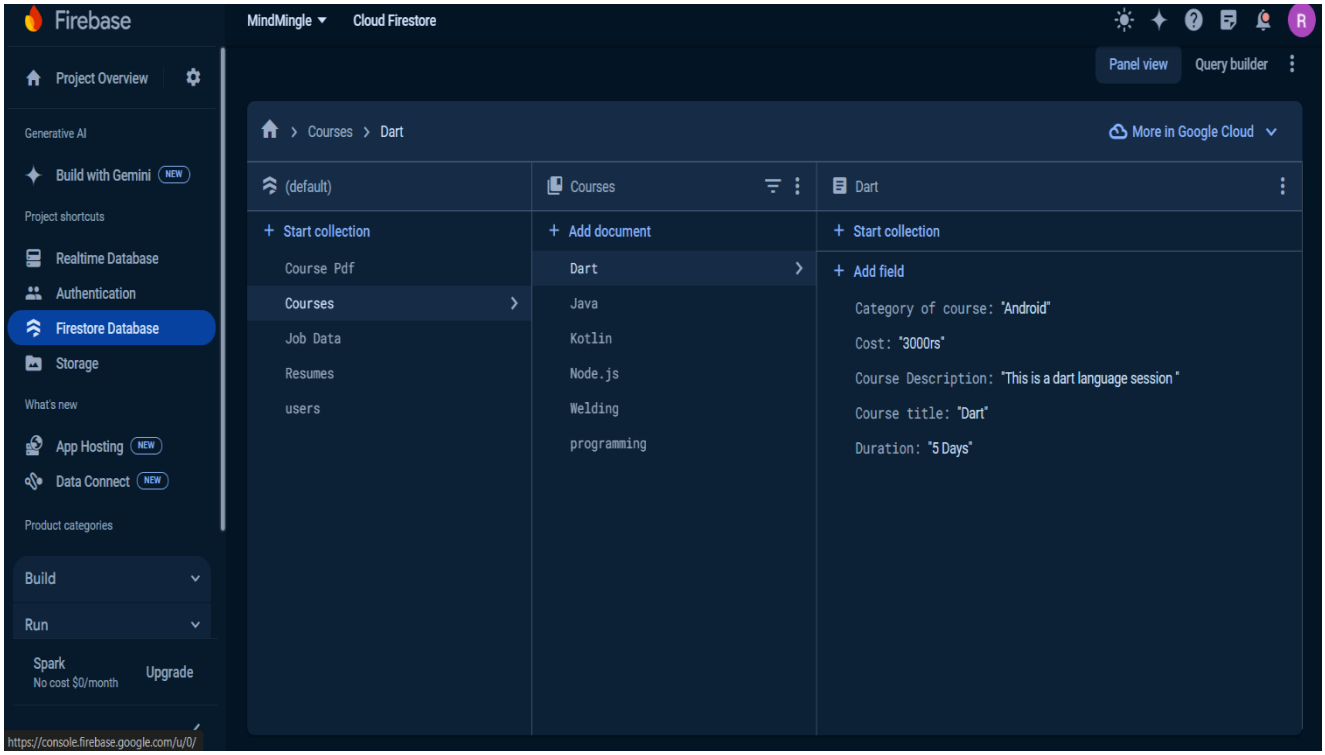


Fig 11 Details of the courses uploaded storage

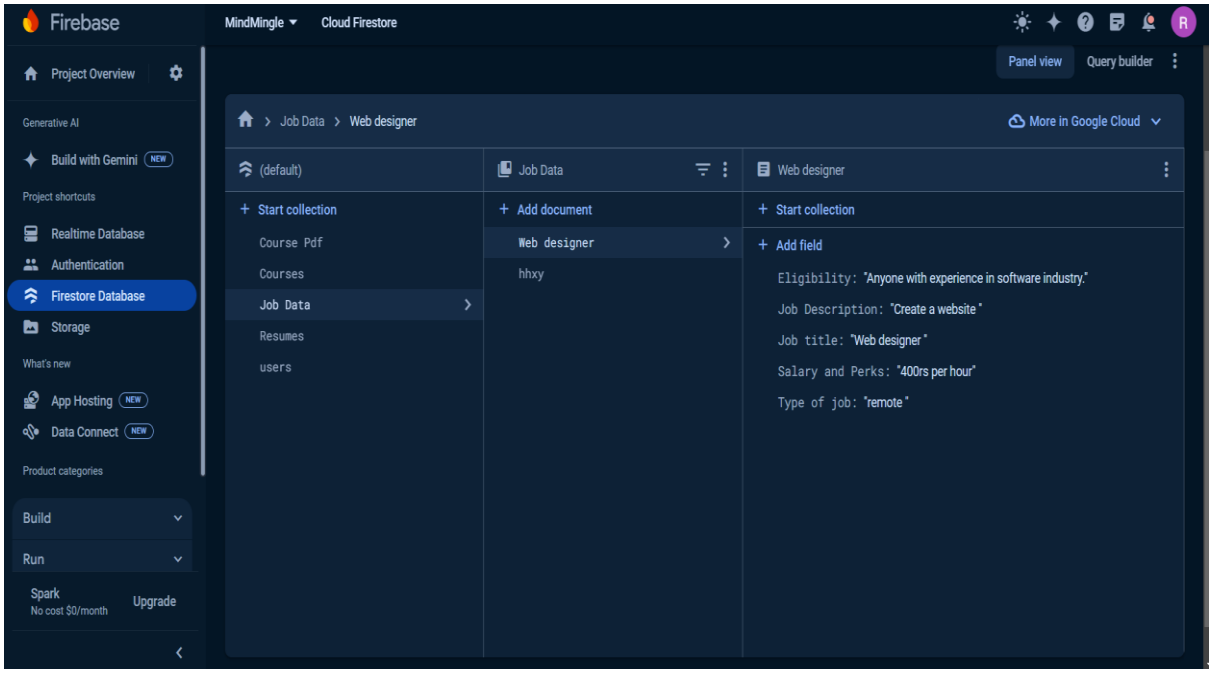


Fig 12 Details of job posted on the app

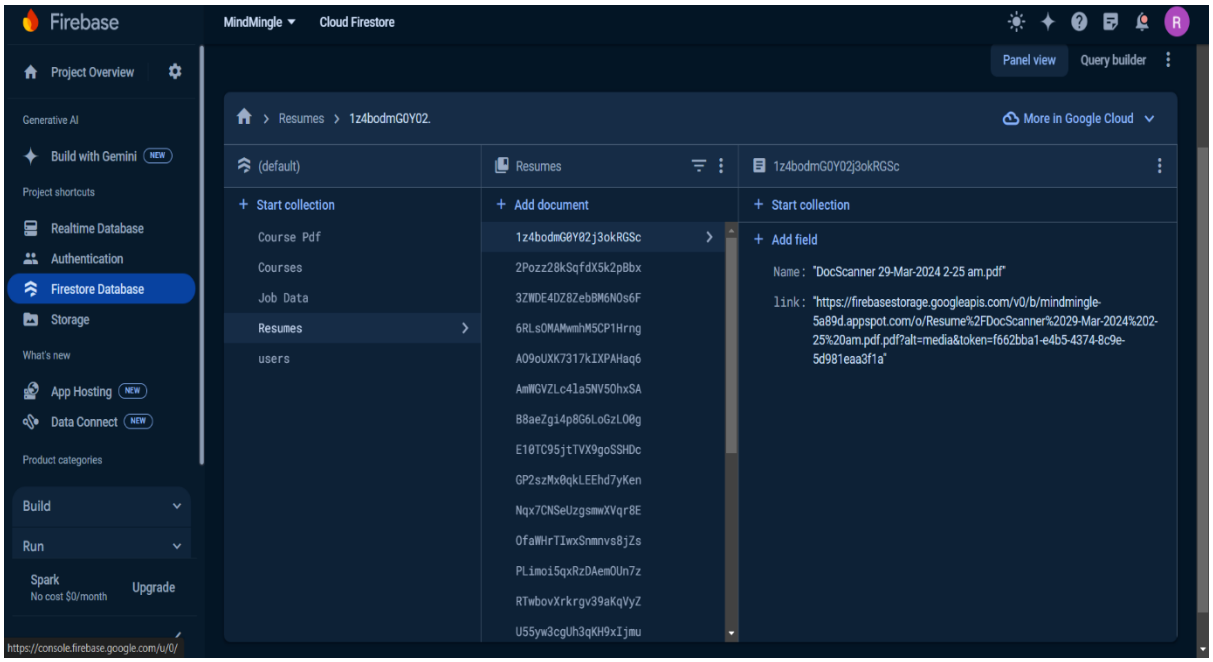


Fig 13 Storage for resumes uploaded on the app

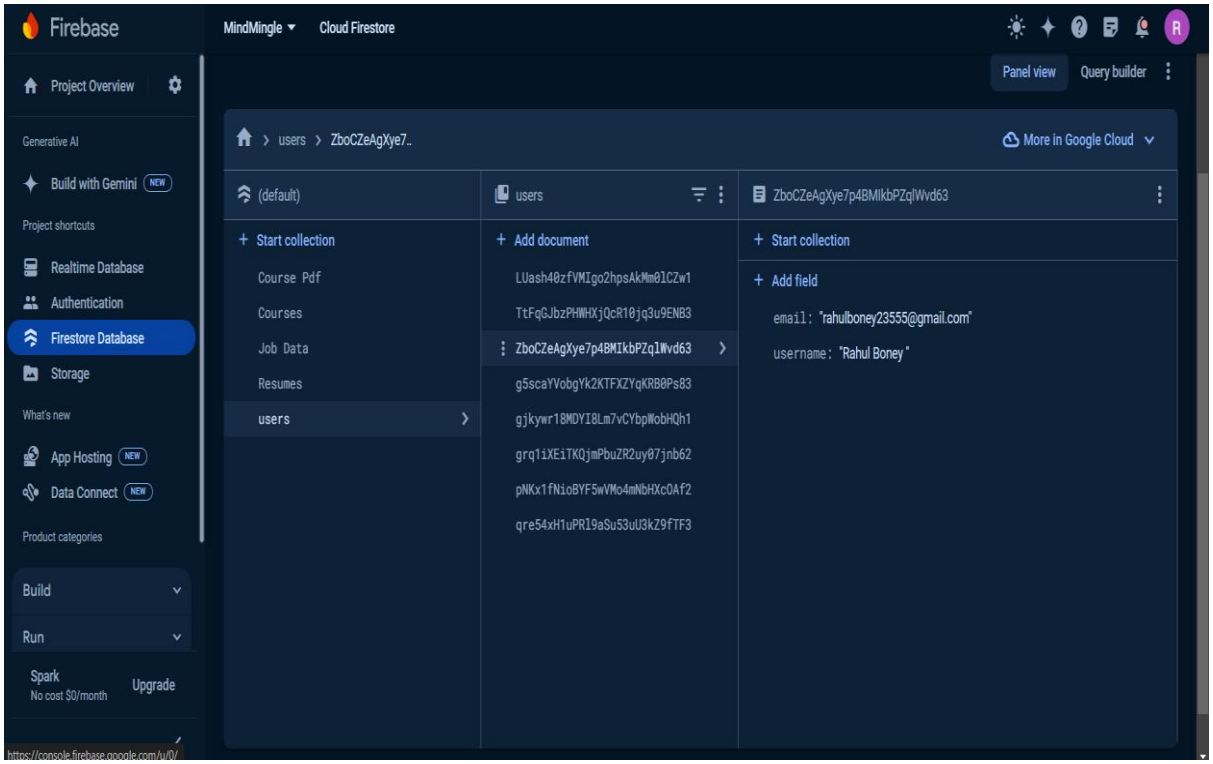


Fig 14 -Details of the users onboarded on the app

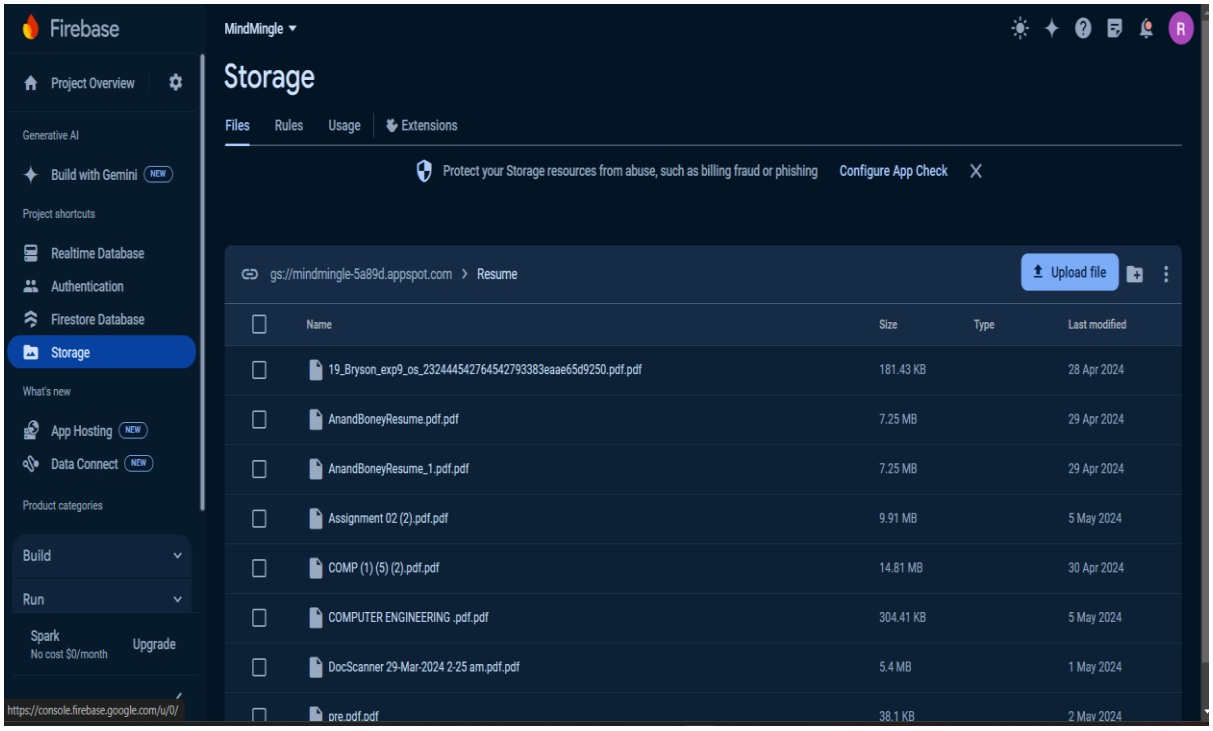


Fig 15 Storage and collected resumes of users on the app

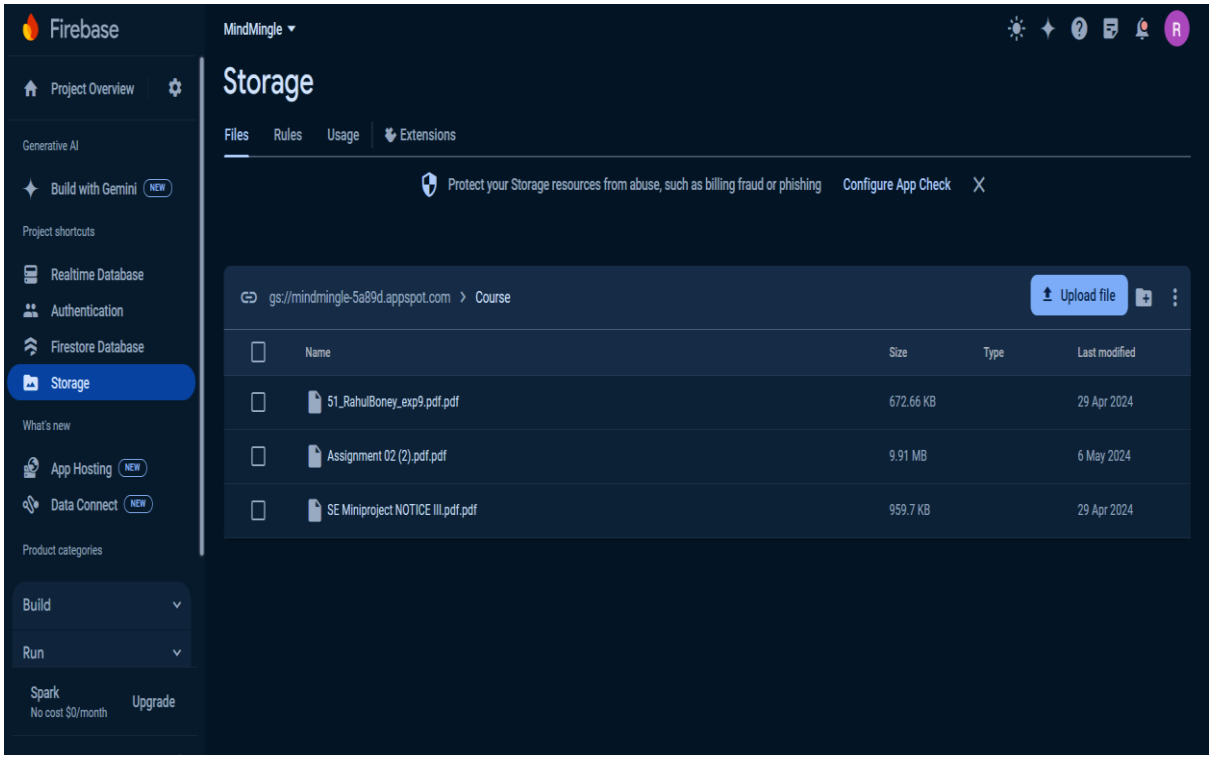
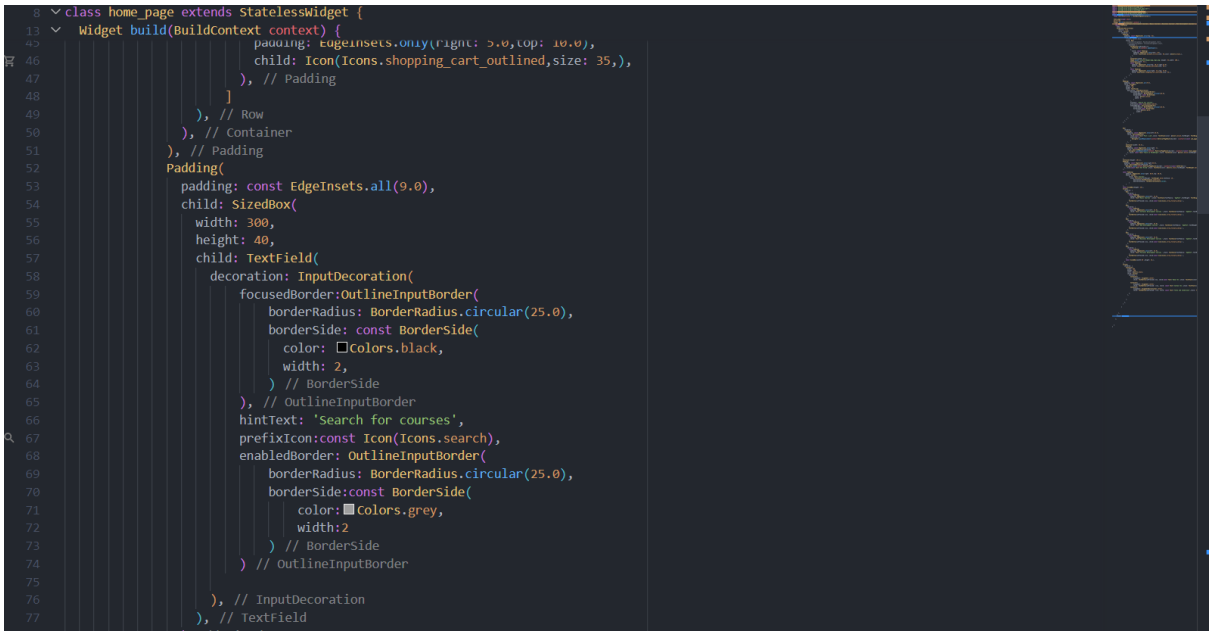


Fig 16 Storage and collection of course structure uploaded by educators on app.



```
class home_page extends StatelessWidget {
  Widget build(BuildContext context) {
                    child: Icon(Icons.shopping_cart_outlined,size: 35,),
                  ), // Padding
                ]
              ), // Row
            ), // Container
          ), // Padding
          Padding(
            padding: const EdgeInsets.all(9.0),
            child: SizedBox(
              width: 300,
              height: 40,
              child: TextField(
                decoration: InputDecoration(
                    focusedBorder:OutlineInputBorder(
                        borderRadius: BorderRadius.circular(25.0),
                        borderSide: const BorderSide(
                          color: Colors.black,
                          width: 2,
                        ) // BorderSide
                    ), // OutlineInputBorder
                    hintText: 'Search for courses',
                    prefixIcon:const Icon(Icons.search),
                    enabledBorder: OutlineInputBorder(
                        borderRadius: BorderRadius.circular(25.0),
                        borderSide:const BorderSide(
                            color:Colors.grey,
                            width:2
                        ) // BorderSide
                    ) // OutlineInputBorder

                ), // InputDecoration
              ), // TextField
```

Fig 17 Sample Code snippet