Glass Ceiling & Co.
Technical Documentation

Group #3
Margie Ruffin, Zari McFadden, Jaida Langham, Lelia Hampton, and Ulunma Egwim

Table of Contents

Overview

This document serves as the technical documentation for the Glass Ceiling web application. It provides information on the technical implementation of the use cases within the Glass Ceiling website. It also goes into detail on the architectural structure of the software. The purpose of this document is to inform the reader how the product works from a technical perspective.

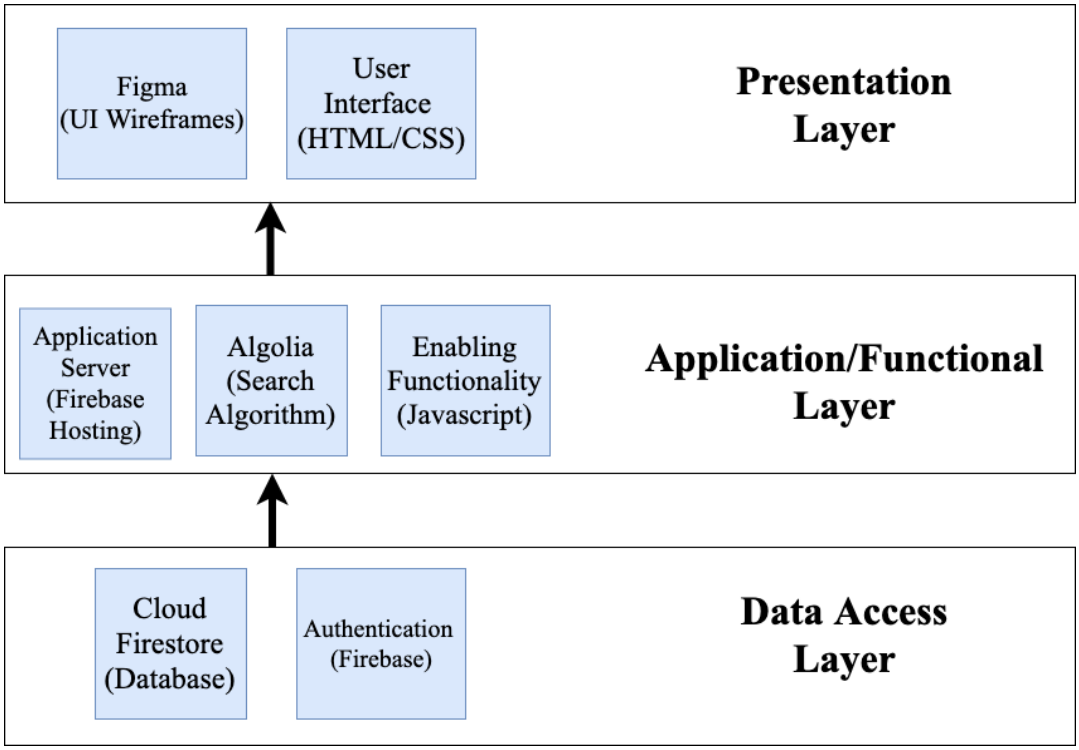
Use and Functionality

Table 1: Use Cases and Their Technical Implementation in the Software

Use Case	Technical Implementation
Employee Register for Glass Ceiling	When the user registers for glass ceiling, they begin on the home page (index.html/index.css) and then they navigate to the "Sign up/Log in" page which is another tab. At which point, they have the option to press the "sign up". They are then redirected to the signup page for candidates. When they sign up their login information is stored in the Firebase Authentication system and an email is sent to them to verify their account.
Employee Log in to Glass Ceiling	When the user registers for glass ceiling, they begin on the home page (index.html/index.css) and then they navigate to the "Sign up/Log in" page which is another tab. At which point, they have the option to press the "log in". They are then redirected to the signup page for candidates. When they login, the Firebase Authentication process occurs.
Complete an Employee Profile	The employee completes a profile once they are registered authenticated using the Firebase Authentication process. Once they input the information they want on their profile into the HTML. This information is then stored in the Cloud Firestore using both Java and JavaScript files.
Search for Job Posting, Job Type, Job Location, or Company	Once logged in, the candidate can search for a job using the Algolia search engine API which will create rankings and allow the user to search by filters and relevance.
Employee Post Review	The candidates will be able to post reviews on the site. The reviews will be input into an HTML form and then put into the Cloud Firestore using Javascript code. It will also be placed onto the website statically.
Employee View Different Page/Tabs	From any given page the user is on, the menu page is visible. At all times, the user has the option to go to the home page, the mission page, and the unconscious bias training page.
Favorite a Job	The candidates will be allowed to favorite a job by clicking on a star icon. Clicking on the star icon will execute JavaScript code which sends the favorited job to the Cloud Firestore to be associated with the candidate and retrieved for viewing on their profile.

Architecture

Figure 1: The Architecture of the Glass Ceiling Web App



Presentation Layer

Figma Wireframes

Figma is being used to help create a visual blueprint of the user interface of our site. We wanted a lightweight software that is available online from a web browser. Figma also allows for collaboration and feedback on wireframes which is important to the development of the software and success of the project because it speeds up the workflow of the user interface development.

HTML/CSS

Once the wireframes are created in Figma, the layout is formatted using HTML and visually enhanced using CSS. Multiple .html files are used within the software architecture in order to navigate from page to page and help to format the layout of the site. CSS is used to make the website visually appealing. For the Glass Ceiling UI, a green-blue color scheme has been chosen. Each of the below .html files has a corresponding CSS file in order to enhance the presentation of the page.

Table 2: Mapping of HTML Files to Pages

HTML File	**Page**
index.html	The index.html file corresponds to the home page which has links to the home, mission, bias training, and signup/login pages. The home page also provides the option to for “looking” or “hiring” for a job.
mission.html	The mission.html file corresponds to the mission page.
bias.html	The bias.html file corresponds to the unconscious bias training page.
candidate/signup-login/signup-login.html	The candidate/signup-login/signup-login.html file corresponds to a page which allows the candidate to choose between signing up or logging in to the website.
candidate/signup-login/signup.html	The candidate/signup-login/signup.html file corresponds to the sign up page for the candidate on the website.
candidate/signup-login/login.html	The candidate/signup-login/signup.html file corresponds to the log in page for the candidate on the website.
candidate/user-profile/user.html	The candidate/user-profile/user.html corresponds to the user’s profile on the website.
company/signup-login/signup-login.html	The company/signup-login/signup-login.html file corresponds to a page which allows the company recruiter to choose between signing up or logging in to the website.

company/signup-login/signup.html	The company/signup-login/signup-login.html file corresponds to the sign up page for the company recruiter on the website.
company/signup-login/login.html	The company/signup-login/signup-login.html file corresponds to the log in page for the company recruiter on the website.
company/hiring-landing/landing.html	The company/hiring-landing/landing.html page corresponds to the landing page for the companies which allows them the option to look for candidates or post a job.
search.html	The search.html file corresponds to the search page.
404.html	The 404.html corresponds to an error page on the website that is triggered by Firebase when a file on the website is not found.

Application/Functional Layer

Application Server (Firebase Hosting)

Firebase Hosting is used to host the website because we are using it for our data storage and it provides fast and secure hosting for web applications. It efficiently serves both our static and dynamic content. It is also free to host on Firebase, and we do not have to allocate another server, lowering costs of software development.

Algolia (Search Engine API)

Algolia is a powerful search engine API. We use it in the software to prepare data as well as send and update data; the data in particular is candidate data on the company recruiter's end and company job data on the candidate's end. The API allows us to manage indices and manage results using relevance, refining, rules, and optimization based on user (candidate/company recruiter) search data. The search engine is secure and uses API keys.

Enabling Functionality (Java and JavaScript)

The Firebase Cloudstore will be communicated with using JavaScript and Java, and the Algolia search engine API will be communicated with using Java. Java is also used to provide functionality when clicking certain buttons on the site that are not navigation based (.html files handle this), save the authentication process which is handled using Firebase's authentication API.

Data Access Layer

Cloud Firestore (Database)

The Cloud Firestore is used for persistent data storage, namely storage of authentication data, user data, company data, company recruiter data, job data, and review data. In the database, these appear as “collections”. The data is taken from the web form and put into the database with Java and JavaScript programs. The data is retrieved from the database using Java programs only.

Authentication (Firebase)

User authentication is done through Firebase since Firebase already has security measures in place such as encryption and decryption. It also handles sending user’s emails for account creations and verifications.