Busybee

Requirements

Specification

# Table of Contents

[**Team Member Names 3**](#_r7zplnnm0h0r)

[**Abstract 3**](#_1uu1tyrsycg0)

[**Tools and Technologies 3**](#_q48ug3ej0k7f)

[**Requirements List 4**](#_k6xn1tt3exi7)

[1. Login Form 4](#_y3jd60jerrs5)

[2. Registration Form 5](#_tn91yjjsxeos)

[3. Dashboard 6](#_dckyisgj2i67)

[4. Dashboard - Home page 7](#_zc9heofpemth)

[5. Dashboard - Jobs Page 7](#_uo99lf8bxp9x)

[6. Dashboard - Statistics Page 9](#_cvewbucmaxvq)

[7. Dashboard - Documents Page 9](#_5tymxy5dagux)

[8. Dashboard - Groups Page 10](#_x5tp0agctbmc)

[9. Database 10](#_e5m9yibay6nh)

[10. Storage 11](#_n9hdv65y10rv)

[**Updated Timeline 12**](#_j6htni91gbq6)

# Team Member Names

Mason Luna, Mason Krause, Jacqueline Justice, Jason Moss

# Abstract

This project will improve the job-seeking experience by providing a dashboard where job seekers can monitor and manage their job applications. Many platforms are involved when looking for a job (Indeed, LinkedIn, Monster, Glassdoor, etc.), and a job seeker is likely using some or all of them. Our project will help users track all the applications they've completed across platforms in one consolidated list. They can keep that list up to date as the statuses of applications change.

The dashboard will be divided into a few main sections. One section will allow the user to store a copy of their resume and cover letter. Another section will contain all applications submitted throughout the job search, with a filtering/sorting mechanism to organize applications as desired (by company, date applied, etc). The final section will show some statistics about the current job search, such as the number of applications per month, the application-to-interview ratio, and the interview-to-offer ratio. These tools will empower users to structure their job search in a deliberate, efficient, and data-driven way.

# Tools and Technologies

Chrome (developer tools), VSCode, React.js, HTML, CSS, JavaScript, PostgreSQL, Vercel, Supabase, Chart.js, Google Fonts, Discord

# Requirements List

## Login Form

### Header text for ‘login’/Login Section with Name of application/logo

### Text box for email

### Text box for password

#### Text in the password field will be masked using bullets.

### Invalid Email/Password Text Label

#### If the user was denied access to the dashboard due to an invalid email or password, a text label will appear and display (Email or Password invalid) in red letters (See [Account Validation](#_gvbr0jnvohp7)).

### Login Button

#### Input Validation: Email

* + - 1. The Email text box must NOT be empty
      2. HTML attributes will be used to ensure a valid email is entered

#### Input Validation: Password

* + - 1. The Password text box must NOT be empty
      2. Passwords must contain 8-18 characters, at least one number, and at least one capital letter.
      3. Password character limit is 18 characters

#### Account Validation

* + - 1. The user’s entered email and password will be validated as a user account using Supabase Auth.

#### Post-login Navigation

* + - 1. Once the user’s account information is validated, they will be navigated to the Dashboard’s homepage   
         (see [Dashboard - Homepage](#_zc9heofpemth)).

### Forgotten Password Link

#### A link with the text “Forgot Password?” will be present and navigate the user to the account recovery page

#### The account recovery page will have a text field for users to enter their email and reset their password.

### Sign Up Now Link

#### A link labeled “Sign up Now” will navigate to the registration form (See [Registration Form](#_tn91yjjsxeos))

## Registration Form

### Header text ‘Register for BusyBee’

### Text input box for first name

### Text input box for last name

### Text input box for email

### Text input box for password

### Text input box for confirming password

### Submit button

* + 1. On click, validate/sanitize input (see [1.5.1](#_eh14hcyggu5) and [1.5.2](#_ohpks9xxeuzr))
       1. If input is valid, send data to the Supabase server.
          1. If the user is already registered, redirect to the login form with an alert that the account already exists with that email
          2. If the user is not already registered, add them into the database, redirect them to the login form with an alert that the account was created successfully
       2. If input is invalid, clear the invalid field and alert the user that a valid input is required for that field.
       3. If multiple fields are invalid, alert the user to the uppermost invalid field.

### Already have an account? Link

* + 1. Link labeled ‘Already have an account? login instead’ will navigate to the Login Form (See [Login Form](#_y3jd60jerrs5))

## Dashboard

### Header

* + 1. A Busybee logo that navigates to the [Dashboard - Homepage](#_zc9heofpemth) on user click
    2. A label with the text “Welcome,” followed by the user’s first name
    3. Sign Out button that signs the user out and navigates to the [Login Form](#_y3jd60jerrs5) on user click

### Footer

* + 1. A Busybee logo that navigates to the Dashboard - Homepage on user click
    2. Resource Section
       1. Static Contact Us Page link
       2. Static FAQ Page link
       3. Static About the Team Page link

### Easy Navigation Section (Left-Hand side)

* + 1. Contains a vertical list of navigation buttons
    2. All buttons will change the Page View into the selected dashboard page on user click.
    3. Buttons, in order
       1. [Jobs Page](#_y99h5j5v61nf)
       2. [Statistics Page](#_o56q4rapzosq)
       3. [Documents Page](#_vesbg9bz4fgl)
       4. [Groups Page](#_c26df1sxs9fn)
    4. Page View
       1. The page view will be implemented according to the requirements of the currently selected dashboard page (Jobs, Statistics, Documents, or Groups)

## Dashboard - Home page

### Statistics Quick View

* + 1. Contains a preview of the statistics based on the most recently added data.
    2. On user click, Statistics Page will open   
       (See [Dashboard - Statistics](#_o56q4rapzosq))

### Jobs Quick View

* + 1. Contains a preview of the most recent jobs that the user has added
    2. On user click, Jobs Page will open   
       (See [Dashboard - Jobs](#_y99h5j5v61nf))

### Documents Quick View

* + 1. Contains a preview of the user’s most recently uploaded documents
    2. On user click, Documents Page will open   
       (See [Dashboard - Documents](#_vesbg9bz4fgl))

### Groups Quick View

* + 1. Contains a preview of the user’s Groups at the bottom of the page
    2. On user click, Groups Page will open   
       (See [Dashboard - Groups](#_c26df1sxs9fn))

## Dashboard - Jobs Page

### Jobs Information Section

* + 1. Contains a count of all jobs added displayed in the top right of the Jobs Page
    2. Add job button
       1. Contains a plus sign and label with “Add a job…”
       2. On click, a form will pop-up for the user to input all information related to a job entry (See [Entities: Job](#_rafjblkoa4n6) 9.1.1.3.2 - 9.1.1.3.12)
          1. Company Name (text box)
          2. Job Title (text box)
          3. Remote or not (check box)
          4. Country (drop-down list)
          5. State (drop-down list)
          6. City (text box)
          7. Date Posted (date picker)
          8. Platform (text box)
          9. Estimated Salary (text box)
       3. Save Button that, on user click, will save the information as a Job Entry and return to the Jobs Page

### Jobs List

* + 1. Displays all recently added or edited jobs that meet the user’s filter requirements with the following information
       1. Job Title
       2. Date Applied
       3. Current application status
    2. Each job in the job list is selectable, and will display the Job Popup once selected

### Job Popup

* + 1. Displays all information stored in the selected job’s entry (Company Name, Job Title, remote, city, state, country, date posted, date applied, platform, estimated salary, application status, company notes, interview notes)
    2. Edit Button
       1. Users will be able to edit all information in the selected job’s entry
       2. The edit button will be replaced by a save button that the users must press to save any changes made
    3. Delete Button
       1. on click, displays a confirmation popup, asking if the user is sure with a button labeled ‘Yes’ and a button labeled ‘No’
          1. When ‘Yes’ is clicked, the job entry is deleted permanently and the popup closes.
          2. When ‘No’ is clicked, the confirmation popup closes.

### Filter and Sort

* + 1. Users will be able to filter the jobs list based on month applied, application status or salary range, where all job entries not matching the criteria will not be shown
    2. Users will be able to change how the job entries are sorted, based on either date applied or alphabetically.

## Dashboard - Statistics Page

### Statistics Graphics

* + 1. Displays the following statistics in square graphics in the body of the page

#### Total number of jobs applied to

#### Applications Submitted this month

#### Average Applications submitted per month

#### Application-to-Interview Ratio

#### Application-to-Offer Ratio

#### Interview-to-Offer Ratio

## Dashboard - Documents Page

### Add Documents Button

* + 1. Contains a plus sign and label with “Add documents…”
       1. On user click, a filepicker will popup and allow user to upload PDF or DOCX files and add it to their documents

### Documents List

* + 1. All documents the user has added will be listed ordered by how recently they were uploaded.
    2. All documents will have an ‘X’ symbol on the far right of the listing for users to delete documents
       1. On click, a confirmation popup will appear and ask if the user is sure with a ‘Yes’ button and a ‘No’ button.
          1. If the user clicks ‘Yes’, the document is deleted and the popup closes.
          2. If the user clicks ‘No’, the popup closes

## Dashboard - Groups Page

#### Tab-based component for Groups

* + 1. Displays selectable name labels of all the user’s created groups
    2. The selected tab will be highlighted similar to Chrome tabs or Microsoft Excel tabs
    3. A button with a plus symbol will be present at the end of the list of tabs to add a new group
       1. On click, a popup will appear for adding a new group with the following information
          1. Group Name (text box)
          2. Group Start Date (text box)
          3. Group End Date (text box) (optional)
       2. When the user is finished inputting the required information, they can click a button labeled ‘Add’ at the bottom of the popup to add the newly created group to their groups.
    4. Any job that has been previously added can be added to any group.

## Database

### Structured Database

* + 1. Entities
       1. User
          1. user\_id : uuid (Primary Key)
          2. first\_name : string
          3. last\_name : string
          4. email: string
          5. password: string
       2. Group
          1. group\_id : int (Primary Key)
          2. group\_name : string
          3. group\_start\_date : date
          4. group\_end\_date : date (NULLABLE)
          5. jobs : list[Job]
       3. Job
          1. job\_id : int (Primary Key)
          2. company\_name : string
          3. job\_title : string
          4. remote: boolean
          5. job\_city: string
          6. job\_state: string
          7. job\_country: string
          8. date\_posted : date
          9. date\_applied : date
          10. platform : string
          11. estimated\_annual\_salary : integer (NULL)
          12. status : string (categorical)
          13. notes : string

## Storage

### Uploaded .PDF and .DOCX files will be stored in Supabase Storage.

* + 1. Add document buttons will be available in the dashboard section “Documents” and on each Job entry
       1. On click, a file selection window will pop up so the file can be uploaded
    2. Replace document buttons will be available in the dashboard section “Documents” and on each Job entry if a file was previously uploaded
       1. The newly uploaded document will replace the old one in storage, the old one will be deleted.

# 

# Updated Timeline

| 2/10 - 2/16 | Mason Krause:   * Plan and develop database schema * Finalize Entity Relationship Diagram   Mason Luna:   * Design project repository structure * Build basic application   Jason Moss:   * Plan React component hierarchy for UI   Jacqueline Justice:   * Set up React with Git Repository |
| --- | --- |
| 2/17 - 2/23 | Mason Krause   * Implement PostgreSQL database * Insert test data into database * Configure Authentication   Mason Luna   * Develop server entry point * Develop continuous deployment in Vercel using Github * Begin defining data models for the application (User, Job, Group, etc)   Jason Moss   * Develop stubs for static papges   Jacqueline Justice   * Registration/Login Form UI |
| 2/24 - 3/02 | Mason Krause   * Configure Supabase backend for request handling   Mason Luna   * Develop RESTful API * Finish defining data models   Jason Moss   * Develop Registration/Login business logic   Jacqueline Justice   * Main Dashboard UI |
| 3/03 - 3/9 | Mason Krause   * Implement Row Level Security for database * Unit test backend processes   Mason Luna   * Set up HTTP client for API communication * Add error handling of HTTP responses   Jason Moss   * Develop Main Dashboard page navigation   Jacqueline Justice   * Job Dashboard UI |
| 3/10 - 3/16 | Mason Krause   * Configure HTTP request handling * Implement stored procedures for predefined queries * Finish unit testing backend   Mason Luna   * Unit Test HTTP requests/responses * Begin integrating database   Jason Moss   * Implement Job Dashboard components   Jacqueline Justice   * Statistic Dashboard UI * Visual on what data is shown * Link buttons to pages |
| 3/17 - 3/23 | Mason Krause   * Configure Supabase storage for documents * Begin integration testing for user flows through UI to backend   Mason Luna   * Implement sorting of jobs on job page * Finish database integration   Jason Moss   * Implement statistics page business logic   Jacqueline Justice   * Job Group UI * Adding new groups * Viewing added groups * Editing groups * Deleting groups * Adding Jobs to groups |
| 3/24 - 3/30 | Mason Krause   * Continue integration testing * Assist with functionality implementation   Mason Luna   * User Input Validation * Finish all Main Dashboard core features   Jason Moss   * Unit Test dashboard components   Jacqueline Justice   * Document Dashboard UI * Uploading PDFs and DOCX files * Viewing added documents * Deleting added documents |
| 3/31 - 4/6 | Mason Krause   * Continue integration testing for all user flows * Test hosting in a production environment   Mason Luna   * Finish all Groups Page core features   Jason Moss   * Implement navigation between Dashboard pages   Jacqueline Justice   * Static Pages FAQ, About, Contact pages |
| 4/7 - 4/13 | Mason Krause   * Finish integration testing * Assist with last minute implementation of features * Ensure hosting configuration is ready for production   Mason Luna   * Finish all Documents page core features   Jason Moss  Jacqueline Justice |
| 4/14 - 4/20 | All: Production deadline for core features |
| 4/21 - 4/26 | All: Debugging |
| 4/27 - 5/02 | All: Debugging and preparation for presentations |