BusyBee

The Job Application Tracker

Team

- Jacqueline Justice
- Mason Krause
- Mason Luna
- 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.

Description

What is BusyBee?

BusyBee will be a web-based software application that functions as a job seeker's application management tool. It will be built with a backend hosted by Supabase using the Express.js framework and a frontend built with React supported by Vercel, HTML, and CSS. BusyBee will run on the Google Chrome browser and be supported on both the desktop and mobile versions of Chrome.

Users will be able to create and login to their accounts to access all information related to job seeking they may have previously submitted. All information relevant to a single job application can be viewed in a packaged format as a single job entry within the application. This information will include any uploaded documents like resumes and cover letters, all jobs they have applied for or have shown interest in (including the company's name, office location, job title, and other relevant information found in the job posting), the current status of the application, the date when the job was applied for, the date when the application status was last verified, and notes the user has left regarding the job entry.

When viewing job entries, users can view a compiled list of all their job entries, showing the company name, job title, application status, and other basic information about the job displayed within the dashboard. This compiled list will be sortable by various means, like date of application, application status, company name, etc. Alternatively, users can select a single job entry, where the user can see a page containing all information related to that job entry. Users can, within the detailed view of a job entry, choose to edit or update any portion of the job entry, and it is expected that users will often add to the notes section as their job application status progresses.

In addition to a compiled list of job entries and the ability to upload documents like resumes and cover letters, users will be able to view statistics on their rate of success from application to interview or from interview to job offer. The ability to view these statistics will assist users in identifying their weaknesses as a job candidate since a low application-to-interview rate would indicate a weak resume or cover letter, and a low interview-to-offer rate would indicate the user needs to improve their interviewing skills.

The primary means for users to use BusyBee will be through the dashboard page. The dashboard will store all relevant user documents, display the compiled list of job entries, display both the application to interview and the interview to offer conversion rates, and other statistics relevant to the user, like their average number of applications added per month. Access to this dashboard allows users to, at a glance, gain a feel for how their job-seeking journey is going overall and compare the number of applications they have submitted in the short term. For example, users could compare the number of applications submitted in March to those submitted in February.

Users may have questions or require additional support while using BusyBee, so in addition to the core functionality of the web application, there will be additional pages that users can navigate to; those pages include an About the Team page detailing each team member and what they contribute to the application, a Frequently Asked Questions(FAQ) page where users can find preemptive answers to common questions that users have, and a Contact Us page to ask uncommon questions, give feedback, or report any bugs that may arise during use.

Why Use BusyBee?

BusyBee is at its best when job seekers are trying to keep their work life organized. By keeping track of all the jobs they have applied to, job seekers are more likely to keep up with their past or future applications. Job seekers will also be more likely to update their resumes and cover letters as they serve as a constant reminder on BusyBee's dashboard. These job seekers are more likely to follow up on their applications and interviews with companies since they are now motivated to update an app and list for themselves.

Not only would BusyBee aid in following up on job seekers' applications, but BusyBee also aids in remembering details of that particular application. When job seekers are searching for more jobs to apply for, they will be able to easily cross-reference their job entry dashboard to see if they have already applied for the position. They will also be able to easily see the number of days that have passed since their job application, so they can, at a glance, determine if it is an appropriate time to send some form of follow-up communication. The ability to quickly make this determination fits squarely into BusyBee’s purpose, which is to increase the efficiency of the job-seeking experience.

Also, job seekers that are BusyBee will be given relevant statistics to indicate areas of improvement as a candidate. For example, a job seeker who understands that they have a low application-to-interview ratio from BusyBee's dashboard will have data-driven evidence that they need to improve their cover letter or resume. By contrast, non-users will be forced to judge based on intuition, memory, and feeling.

Who is BusyBee for?

BusyBee can be used by recent graduates applying for as many jobs as possible. They need to keep track of which jobs they have applied for and update their statuses as they receive responses from employers. BusyBee can be used by active job seekers applying to the competitive job market who need to keep track of their applications. BusyBee is excellent for students and early-career job seekers who wish to be organized and stay on top of their job decisions, learn about their personal application statistics, and want to be reminded of their applications and statuses.

Feature List

Critical Features (finished by end of semester)

- Users can create an account or login to their account to view their personal job search and application status dashboard.

- Users can upload a .DOCX or .PDF file of their resume and cover letter to be stored and displayed in a downloadable format on their dashboard. Keeps the user's up-to-date resume and cover letter for easy access.

- Users can create, view, update, and delete potential job entries. Job entries consist of data points like company name, office location, job title, link to job posting, date job was posted, where the listing was found, salary range, updatable application status, date applied, resume submitted, cover letter submitted, notes about the potential job, notes from interview, etc.

- Users can view a dashboard containing a list of all job entries they have created and basic information within those job entries, like company name, job title, link to job posting, and application status.

- Users can select an individual job entry within their dashboard to view the details of a job entry, where they will be able to see and edit all information they have entered that is contained in that job entry.

- Users can view their application to interview conversion rate and interview to offer conversion rate.

- Contact Page, About the Team Page, FAQ page (static pages).

Possible Features (may be added if time allows)

- A progress bar that fills in as a potential job changes statuses and has a celebratory animation when a potential job makes an employment offer.
- An analysis tool that can explain to users where a majority of potential jobs are becoming rejections to help users identify the areas they need most improvement in.
- Email integration system that emails users to follow up on their job applications to the employers and sends reminders about the jobs they have added.
- Rate your interview/application experience page for yourself (possibly others).

Future Features (nice to have but cannot be implemented in time)

- A list of job recommendations from various job board platforms compiled using those platforms' Application Programming Interfaces (APIs) so that a job seeker only needs to check one place for new job postings instead of multiple sites and apps.

Technology

Platform

Web

Browser

Chrome

Operating System

Windows

Integrated Development Environment

VSCode

Programming Languages

Frontend

React.js, HTML, CSS, JavaScript

Backend/Database

PostgreSQL

3rd Party Libraries and Tools

Google Fonts, Supabase, Chart.js

Server Software

Vercel (frontend), Supabase (backend/database)

Communication Software

Discord

Server Information

Vercel (frontend), Supabase (backend/database)

Data Sources

This project does not rely on any outside data source. Users will generate the data used in this app. For development, we will populate the database with custom dummy data to test functionality.

Team Member’s Backgrounds and Responsibilities

Backgrounds

- Jacqueline Justice: 3 years experience with Java, JavaFX. 2 years experience with HTML, CSS, JavaScript. 1 year experience with programming documentation/UML for architecture and requirements.
- Mason Krause: 3 years experience with HTML, CSS, JavaScript and some exposure to frameworks. 2 years experience with structured databases.
- Mason Luna: 1 year experience with JavaScript and Node.js, some exposure to HTML and CSS, no experience with React or Supabase.
- Jason Moss: 1 year experience with HTML web development. Experience with programming in JavaScript along with some Object Oriented Programming (OOP) and experience with Node. No experience with server requests and responses.

Responsibilities

- Jacqueline Justice: User Interface Design and Implementation
- Jason Moss: Frontend Implementation
- Mason Luna: Supabase Configuration, Frontend Implementation
- Mason Krause: Database structure design and implementation, HTTP requests/responses

Dependencies, Limitations, and Risks

Dependencies

- Frontend and Backend Integration: Backend APIs need to be implemented and tested before the frontend can be fully completed. The team will stick to their schedule and work together to implement and test the API and requests before the front end completes their task.

- Hosting through 3rd party services (Vercel and Supabase): To get our app online for the public to engage with, we'll rely on these 3rd party services to deploy and host our system. To mitigate risks, we will thoroughly review the configuration documentation and test the deployment process with mock code in advance, well before deploying our final application, so that any complications that arise can be researched before the deadline.

Limitations

- Time Constraint: 14-15 weeks to develop a fully functional web-based application. The team will stick to their schedule and tasks accordingly and notify the rest of the team if they are falling behind schedule.

- Team Experience: The team members need to learn and develop software skills with new technologies. To overcome these limitations, we will thoroughly review the documentation on the chosen technologies, and then we will learn and develop incrementally, with specific functionality targets in mind in order to minimize extraneous learning and effectively manage our time. In this way, we will develop our knowledge and skills using new technologies as we develop the application.

Risks

- Security: Security is always a risk when handling user data and information. The team will use secure tools provided by Supabase to handle authentication and make sure the users' information is safe and secure.

- 3rd Party Libraries: Using 3rd party libraries can be troublesome if the library is experiencing issues or trouble on their end. The team will wait to see if the library is updated and fixed and if it is not, a new library will be used to ensure the proper data is shown.

Timeline

- ## Week Ending 2/1 - Week 1
  - Design UI layout

    - Finalize must-have features, pages, components
    - Website wireframe discussion and layouts
    - Entity relationship discussion and draft

- ## Week Ending 2/8 - Week 2
    - Non-functional and functional requirements documentation
    - Use case documentation
    - UML diagrams of architecture with functions/libraries
    - Finalize/Agree on wireframes and layouts
    - Finalize entity relationship diagram

- ## Week Ending 2/15 - Week 3
    - Set up GitHub repository
    - Create React App
    - Create Supabase Project
    - Start on main dashboard UI development
    - Implement database schema

- ## Week Ending 2/22 - Week 4
    - Continue main dashboard UI development
    - Start create account/login UI development
    - Configure Supabase backend services
    - Create fake cover letter/resume for testing (DOCX, PDF)
    - Populate database with dummy data

- ## Week Ending 3/1 - Week 5
    - Finish main dashboard UI development
    - Continue create account/login UI development
    - Test connection from React app to Supabase via HTTP request
    - Start account creation/login functionality

- ## Week Ending 3/8 - Week 6
    - Finish create account/login UI
    - Finish account creation/login functionality
    - Unit test account creation/login functionality

    - Start Create/View/Edit/Delete job entry UI
    - Start frontend development of HTTP requests for job entries
    - Implement file upload for resume and cover letter

- ## Week Ending 3/15 - Week 7
    - Continue Create/View/Edit/Delete job entry UI
    - Continue frontend development of HTTP requests for job entries
    - Unit test file upload for resume and cover letter

- ## Week Ending 3/22 - Week 8
    - Finish Create/View/Edit/Delete job entry UI
    - Finish frontend development of HTTP requests for job entries

- ## Week Ending 3/29 - Week 9
    - Unit test Create/View/Edit/Delete job entry functionality
    - Polish main dashboard UI, specifically the job entry list

- ## Week Ending 4/5 - Week 10
    - Implement filtering and sorting functionality of the job entry list
    - Implement statistics calculations such as application to interview ratio, number of applications per month since job search began.
    - Start on rendering statistics with Chart.js on the main dashboard.

- ## Week Ending 4/12 - Week 11
    - Finish charting functionality
    - Unit test filtering/sorting and statistics calculating and charting functionality.
    - Start on supplementary pages (about us, contact us, FAQ)
    - Start user testing amongst the team and gather feedback

- ## Week Ending 4/19 - Week 12
    - Test hosting servers with app prototypes to ensure smooth deployment. Document any issues/complications

    - Change/Update from User testing/feedback
    - Wrap up unit testing
    - Start end-to-end integration testing to ensure all features work seamlessly together

- Week Ending 4/26 - Week 13

    - Update Documentation details
    - Finish unit/integration tests
    - Last minute, final UI tweaks/bug fixes. Final preparations for release

- Week Ending 5/3 - Week 14

    - Deploy application/Final Presentation