SENG2021

Deliverable 5

Group: AT3K

26th April 2021

Summary

This report provides a detailed explanation of our applications business value and design implementation. Each member's personal experience in searching for a job contributed to our final choice of problems to address. This project aims to make managing and applying to jobs into a more efficient process. To fulfil this specification, we decided to focus on creating a clear user interface, creating job boards and being able to track jobs to add to these boards.

The main factors when choosing the technology for each 'level' of our application was ease of use, ease of integration within the system and how suitable it is for our purpose. The chosen technology for our frontend was React JS along with Material UI for its library of components. This front end will communicate with a Flask server via the REST API as it is a system that all team members are familiar with. The function modules will communicate with the MongoDB database along with the Google, CareerJet, Wikipedia and Clearbit APIs. The main design elements of the application are evident through the sequence, UML class and ER diagrams which illustrate our use cases, frontend interactions and data model respectively.

Part I: Business

Introduction to Problem and System:

The collective job searching experience of the team enabled us to identify what current job searching applications are lacking, providing us with the necessary perspective to make and address appropriate requirements. Our core requirement was to create a user friendly and intuitive platform to manage job applications. From this core idea we branched off into addressing specific requirements such as creating a convenient and easy to use UI, being able to track jobs through boards and viewing statistics pertaining to our progress in searching for a job.

Problem Statements:

(1) There is no tool that integrates the job postings from multiple sources with an automated and customisable spreadsheet/board interface for tracking the application process for those jobs.

(2) There is a lack of job tracking applications that focus on providing free user analytics and showing summaries such as the rate of response for applications or the interview success percentage.

(3) Existing tools do not have very user-friendly interfaces or sufficient guidelines/tutorials for beginners of the job-hunting process.

(4) Major job-hunting platforms do not provide many ways to toggle the level of detail for the lists of job postings.

Main Features:

The main features of our application aim to directly address our problem statements.
List of different features of your system.

Bullet point list of main features + a bit of context behind those features if necessary

- Being able to create a job board for collating all jobs that you want to track. Boards can have multiple views based on what information you need.
    - Spreadsheet view: View all the information about tracked jobs in a spreadsheet. You may also pick and choose which fields you want displayed based on preference.
    - Kanban view: Easily see that states that different tracked jobs are in such as resume sent, or interview scheduled.
    - Calendar view: See on a calendar any key dates such as interview dates or application deadlines.
- Viewing statistics relevant to your job searching progress.
    - Line chart showing user activity across a selectable time range for a specific job board.
    - Visualising the percentage of jobs that have been tracked, applied for, interviewed for and finalised.
    - Can see the percentages of responses out of resumes sent.
- Searching for jobs by title and location
    - Toggling the level of detail in job postings during job search.
        - Details of the job posting can be adjusted based on user preference.
    - Job postings can be clicked on leading to a page with extensive information about that job
        - The company's profile page can be accessed by clicking on the company title on the job details page

User Stories:

Job Sorting Strategy:

Feature: Ordering the search results by posted date

As a job seeker
So that I can find a job that has been posted recently
I want to sort the search results by the date posted.

Scenario: Sort the search results by the date posted from earliest to latest
GIVEN that I am on the "Search For Jobs" page
WHEN I click on any category or enter a search query
THEN I should see a list of job results
WHEN I click on "Sort by"
THEN I should be able to see a list of sorting strategies, such as "Posted Date (most recent to least recent)".

Toggleable Level of Detail:

Feature: Providing more detailed information for each job card

As a job seeker
So that I can quickly browse the search results without clicking each job to see its details
I want to view full title, posted date and the company of each job, and optionally view additional details

Scenario: View job search results in more detail
GIVEN that I am on the "Search For Jobs" page
WHEN I click on "Job search" search bar and input some keywords
THEN A list of job cards are displayed with title, company name, salary, location and link to original job post (from Careerjet, etc.)
WHEN I click on the "Toggle detail" dropdown
THEN it should show options for different levels of detail that I can choose to display
WHEN I click on "More detail"
THEN the search results should display fields such as job title, company name, link to original job post, salary, location, posted date, and a brief description
WHEN I click on "Customise Fields" button
THEN I see a list of additional toggles for fields to be displayed
WHEN I disable a field by clicking on the toggle for the field
THEN the job posts no longer displays the disabled field

Board Management:

Feature: Creating new job boards

As a job seeker
So that organise the job applications I am interested in tracking
I want to be able to create new boards of job postings

Scenario: Create new job boards for tracking job applications
GIVEN that I am on the "Job Dashboard" page
WHEN I click the "Create New Board" button
THEN a modal containing a form should appear
AND I should be able to fill out the fields for creating the new board
WHEN I click the "Create" button
THEN I should see a new job board card on the dashboard

Feature: Managing board descriptions and names

As a job seeker
So that I can prevent the creation of multiple boards for the same requirements
I want to be able to give descriptions to each board

Scenario: Give a board a short description of its purpose
GIVEN that I am on the "Job Dashboard" page
WHEN I click ellipses in the top right corner of the board cards
THEN I should see a dropdown menu to "Edit this board" or "Delete this board"
WHEN I click "Edit this board"
THEN I should be shown a modal popup containing a form
WHEN I click "Save" after having completed the form
THEN I should be able to see the changes reflected on the board card for the board I edited

Automated Job Tracking:

Feature: Saved job postings automatically generates new entries in the Kanban board/spreadsheet for a selected board

As a job seeker
So that I can conveniently manage each of my job applications
I want to track the job postings that I find on the site which interest me

Scenario: Automatically generate a new entry in the Kanban board/spreadsheet for a selected board
GIVEN that I am on the "Search For Jobs" page
WHEN I click on "Job search" search bar and input some keywords
THEN A list of job cards are displayed with title, company name, salary, deadline, location and posted date
WHEN I click on the dropdown 'Selected Board'
AND select the board I want
AND I click on a 'Track' button on a job card
THEN I should see the button text indicate that the posting is now being tracked
AND a new entry is generated in the job board with columns such as 'Company', 'Title', 'Date', 'Status' automatically populated

Toggleable Board Views:

Feature: Allow the user to toggle between different 'views' of the tracked jobs, such as switching between Kanban board, spreadsheet or list

As a job seeker
So that I can manage my job applications in way that is most intuitive to me
I want to be able to toggle between alternative representations of the same list of job opportunities I have saved

Scenario: Change the view from board to spreadsheet
GIVEN that I am on the "Job Dashboard" page
WHEN I click "VIEW" button on any board card
THEN I should see a detailed dashboard of all the job postings I am tracking
WHEN I click a dropdown button with the text 'Board Type'
THEN I should be able to select one of the views from the dropdown menu
AND the job board should change its representation to reflect the option I selected

Saving Favourite Companies:

Feature: Saving my favourite companies in my personal dashboard

As a job seeker
So that I can efficiently manage the company's information that I'm interested in
I want to save companies' profiles to view from my "Dashboard" page

Scenario: Saving my favourite companies from job detail page
GIVEN that I am on the "Job Description" page
THEN I should be able to see a "Favourite Company" button under the job title

WHEN I click the “Favourite Company” button
THEN The company profile is saved to my “Job Dashboard” page
AND I should be able to find it from “Favourite Companies” section in “Job Dashboard” page.

User Analytics Visualisation:

Feature: Data visualisation graphics and text showing the user's activity, application response rate and success rate for interviews

As a job seeker
So that I am informed about my long-term job-hunt progress
I want to view real aggregate statistics about my activity

Scenario: View “All activities” statistics for the past three days for a specific board
GIVEN that I am on the "Statistics” page
THEN I should see dropdown menus with text ‘Last 7 days’, ‘Activity Type’ and ‘Selected Board’
WHEN I click the dropdown ‘Last 7 days’
AND select ‘Last 3 days’
THEN I click the dropdown ‘Activity Type’
AND select ‘All Activities’
THEN I click the dropdown ‘Selected Board’
AND select a specific board
THEN graphs and text summaries are rendered on the page illustrating all activity over the last three days for a specific board

Board Edition:

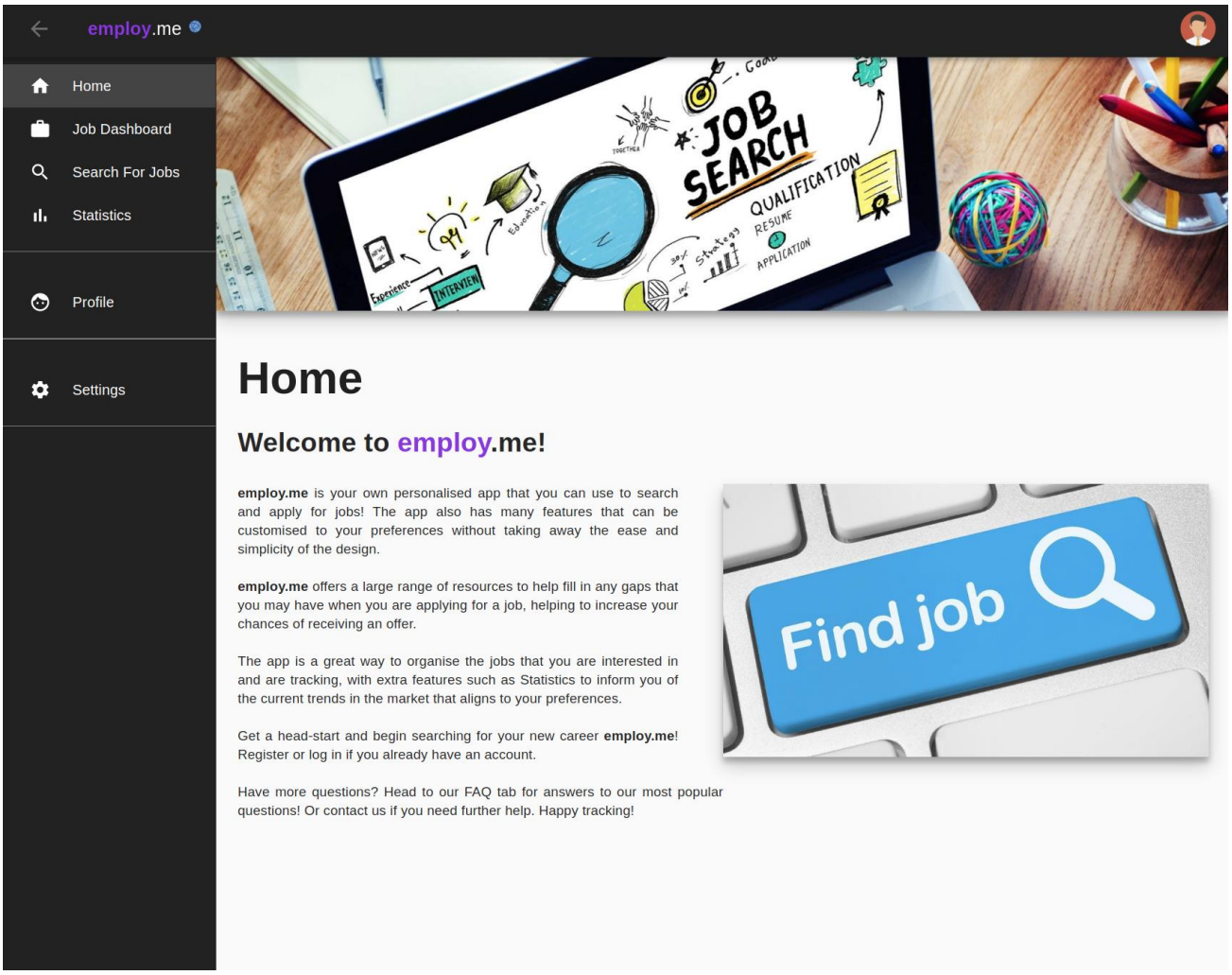
Feature: Allow the user to edit the details of any board they have.

As a job seeker
So that I can control the status or have a reminder of what to do next for the job application
I want to conveniently update the details or add event to a tracked job

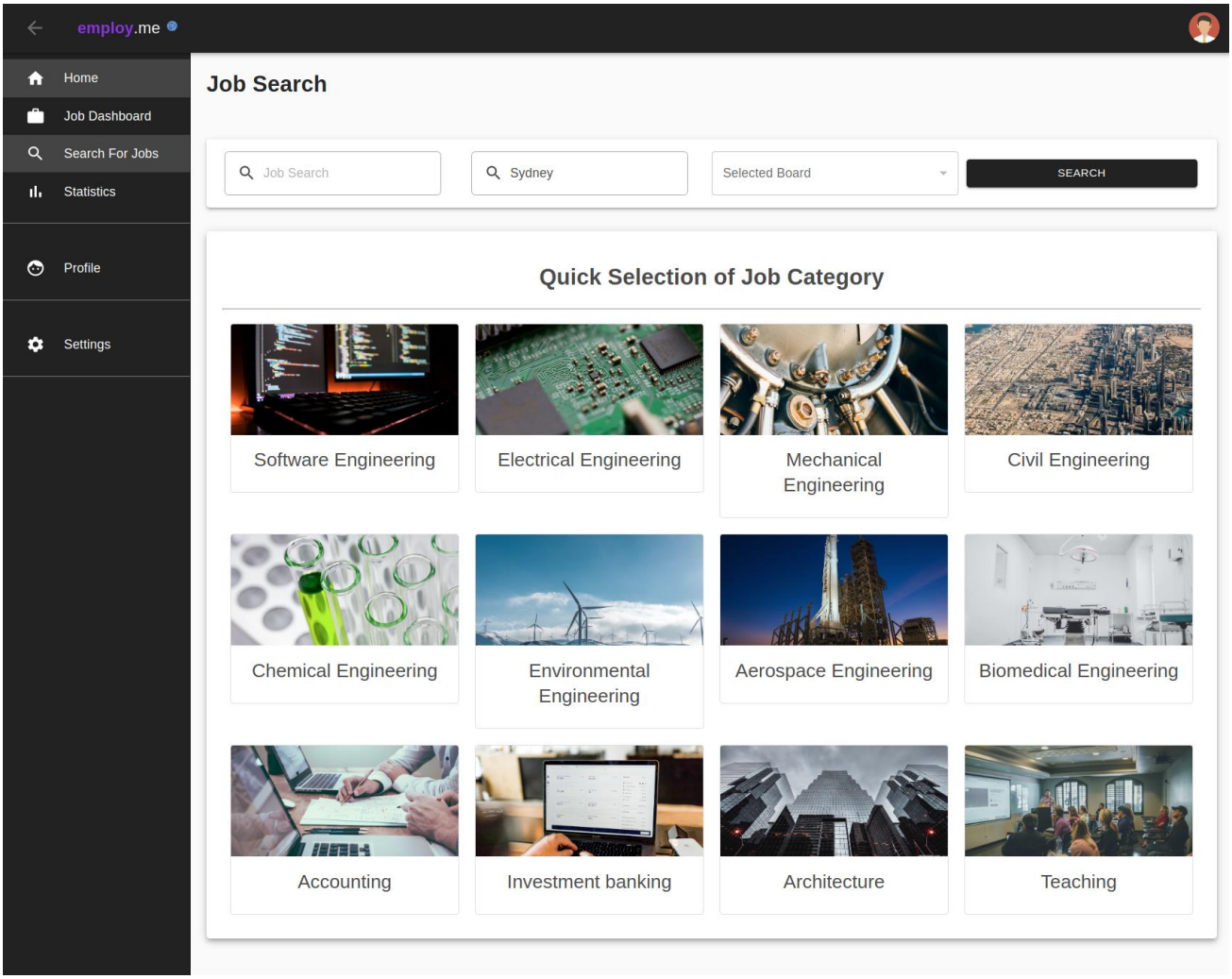
Scenario: Updating a board in spreadsheet version
GIVEN I am on the “Job Dashboard” page
WHEN I click “VIEW” button on any board card
THEN I should see a detailed dashboard of all the job postings I am tracking
WHEN I click “Enter Edit Mode” button from the bottom of the page
THEN all the cells become editable
WHEN I scroll the spreadsheet to the right
THEN I should be able to see an “Events” column
WHEN I enter “Interview” to the “Event Name” text input box on the first row
AND I select 25th April 2021 by clicking the white input box under “Event Name” input box
AND I click “Confirm” button
THEN I should be notified “Created a new event: ‘Interview’ at...” and “Saved Changes”
WHEN I click “Exit Edit Mode” from the bottom of the page
THEN I should be able to see a new event called “Interview” has been added to my first tracked job

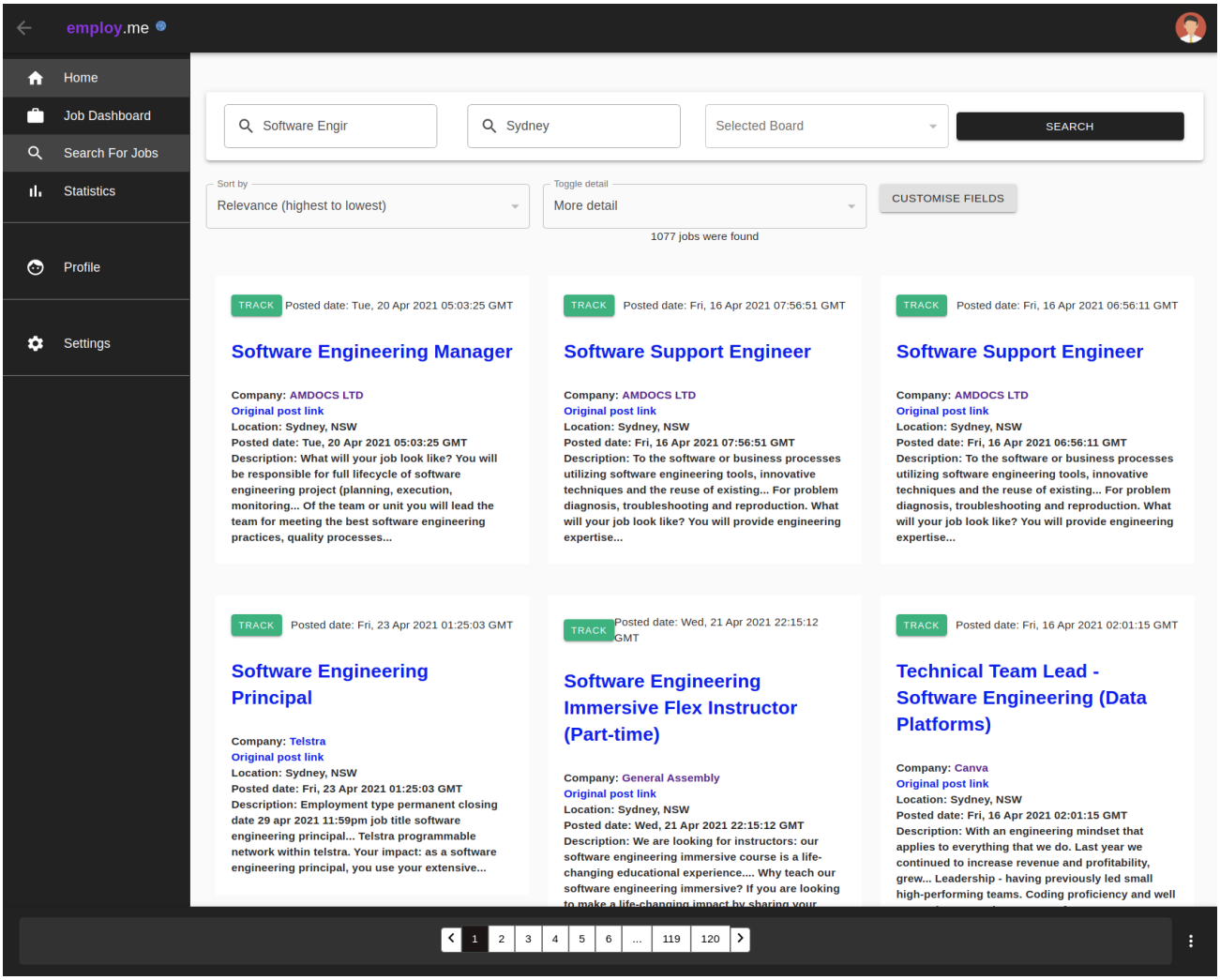
Screenshots:

Home:

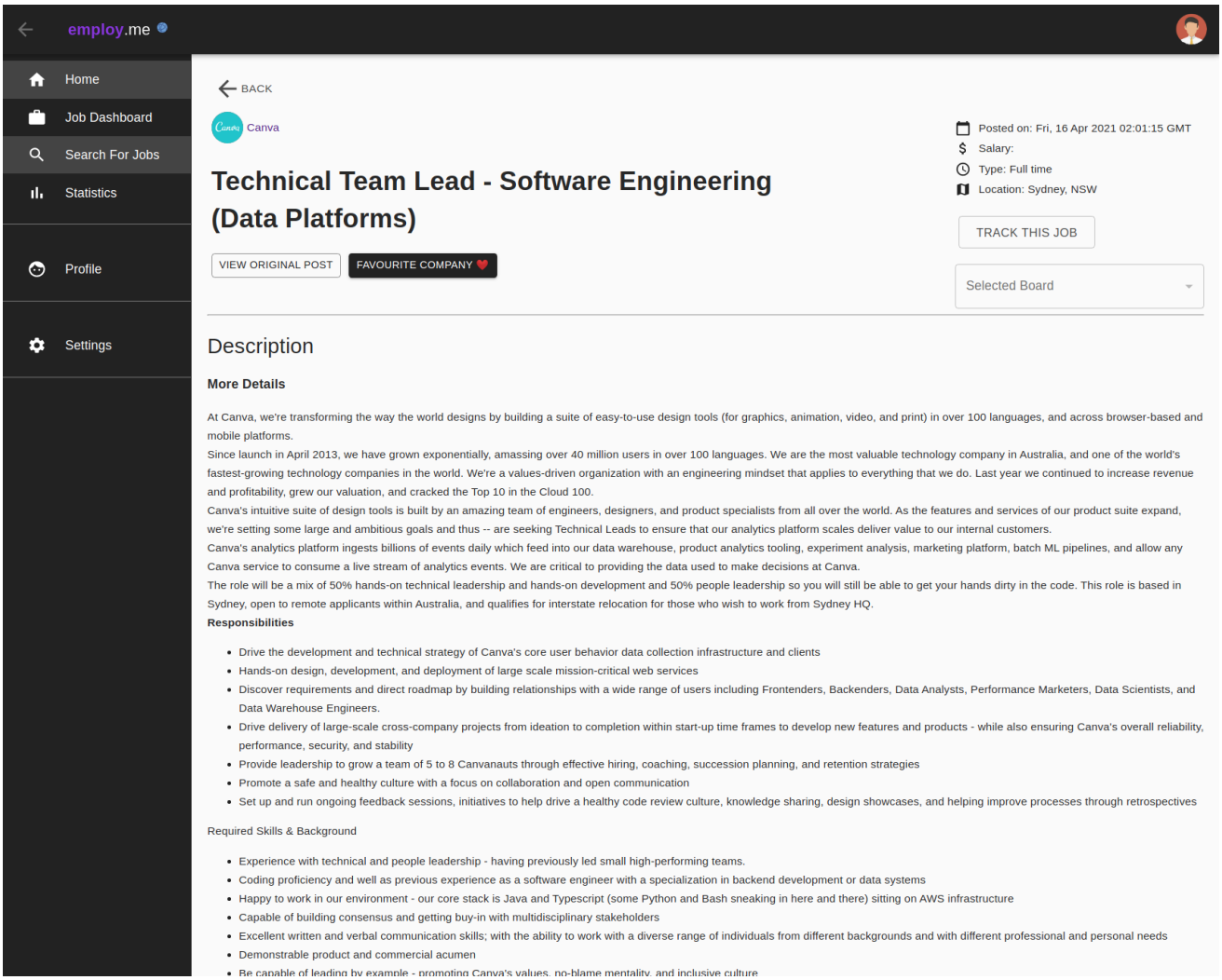


Search:


employ.me
Home
Job Dashboard
Search For Jobs
Statistics
Profile
Settings
Job Search
Job Search
Sydney
Selected Board
SEARCH
Quick Selection of Job Category
Software Engineering
Electrical Engineering
Mechanical Engineering
Civil Engineering
Chemical Engineering
Environmental Engineering
Aerospace Engineering
Biomedical Engineering
Accounting
Investment banking
Architecture
Teaching


employ.me
Home
Job Dashboard
Search For Jobs
Statistics
Profile
Settings
Software Engir
Sydney
Selected Board
SEARCH
Sort by
Relevance (highest to lowest)
Toggle detail
More detail
CUSTOMISE FIELDS
1077 jobs were found
TRACK
Posted date: Tue, 20 Apr 2021 05:03:25 GMT
Software Engineering Manager
Company: AMDOCS LTD
Original post link
Location: Sydney, NSW
Posted date: Tue, 20 Apr 2021 05:03:25 GMT
Description: What will your job look like? You will be responsible for full lifecycle of software engineering project (planning, execution, monitoring... Of the team or unit you will lead the team for meeting the best software engineering practices, quality processes...
TRACK
Posted date: Fri, 16 Apr 2021 07:56:51 GMT
Software Support Engineer
Company: AMDOCS LTD
Original post link
Location: Sydney, NSW
Posted date: Fri, 16 Apr 2021 07:56:51 GMT
Description: To the software or business processes utilizing software engineering tools, innovative techniques and the reuse of existing... For problem diagnosis, troubleshooting and reproduction. What will your job look like? You will provide engineering expertise...
TRACK
Posted date: Fri, 16 Apr 2021 06:56:11 GMT
Software Support Engineer
Company: AMDOCS LTD
Original post link
Location: Sydney, NSW
Posted date: Fri, 16 Apr 2021 06:56:11 GMT
Description: To the software or business processes utilizing software engineering tools, innovative techniques and the reuse of existing... For problem diagnosis, troubleshooting and reproduction. What will your job look like? You will provide engineering expertise...
TRACK
Posted date: Fri, 23 Apr 2021 01:25:03 GMT
Software Engineering Principal
Company: Telstra
Original post link
Location: Sydney, NSW
Posted date: Fri, 23 Apr 2021 01:25:03 GMT
Description: Employment type permanent closing date 29 apr 2021 11:59pm job title software engineering principal... Telstra programmable network within telstra. Your impact: as a software engineering principal, you use your extensive...
TRACK
Posted date: Wed, 21 Apr 2021 22:15:12 GMT
Software Engineering Immersive Flex Instructor (Part-time)
Company: General Assembly
Original post link
Location: Sydney, NSW
Posted date: Wed, 21 Apr 2021 22:15:12 GMT
Description: We are looking for instructors: our software engineering immersive course is a life-changing educational experience.... Why teach our software engineering immersive? If you are looking
TRACK
Posted date: Fri, 16 Apr 2021 02:01:15 GMT
Technical Team Lead - Software Engineering (Data Platforms)
Company: Canva
Original post link
Location: Sydney, NSW
Posted date: Fri, 16 Apr 2021 02:01:15 GMT
Description: With an engineering mindset that applies to everything that we do. Last year we continued to increase revenue and profitability, grew... Leadership - having previously led small high-performing teams. Coding proficiency and well
1 2 3 4 5 6 ... 119 120

Job Details:


employ.me
Home
Job Dashboard
Search For Jobs
Statistics
Profile
Settings
BACK
Canva
Technical Team Lead - Software Engineering (Data Platforms)
VIEW ORIGINAL POST
FAVOURITE COMPANY
Posted on: Fri, 16 Apr 2021 02:01:15 GMT
Salary:
Type: Full time
Location: Sydney, NSW
TRACK THIS JOB
Selected Board
Description
More Details
At Canva, we're transforming the way the world designs by building a suite of easy-to-use design tools (for graphics, animation, video, and print) in over 100 languages, and across browser-based and mobile platforms.
Since launch in April 2013, we have grown exponentially, amassing over 40 million users in over 100 languages. We are the most valuable technology company in Australia, and one of the world's fastest-growing technology companies in the world. We're a values-driven organization with an engineering mindset that applies to everything that we do. Last year we continued to increase revenue and profitability, grew our valuation, and cracked the Top 10 in the Cloud 100.
Canva's intuitive suite of design tools is built by an amazing team of engineers, designers, and product specialists from all over the world. As the features and services of our product suite expand, we're setting some large and ambitious goals and thus -- are seeking Technical Leads to ensure that our analytics platform scales deliver value to our internal customers.
Canva's analytics platform ingests billions of events daily which feed into our data warehouse, product analytics tooling, experiment analysis, marketing platform, batch ML pipelines, and allow any Canva service to consume a live stream of analytics events. We are critical to providing the data used to make decisions at Canva.
The role will be a mix of 50% hands-on technical leadership and hands-on development and 50% people leadership so you will still be able to get your hands dirty in the code. This role is based in Sydney, open to remote applicants within Australia, and qualifies for interstate relocation for those who wish to work from Sydney HQ.
Responsibilities
Drive the development and technical strategy of Canva's core user behavior data collection infrastructure and clients
Hands-on design, development, and deployment of large scale mission-critical web services
Discover requirements and direct roadmap by building relationships with a wide range of users including Frontenders, Backenders, Data Analysts, Performance Marketers, Data Scientists, and Data Warehouse Engineers.
Drive delivery of large-scale cross-company projects from ideation to completion within start-up time frames to develop new features and products - while also ensuring Canva's overall reliability, performance, security, and stability
Provide leadership to grow a team of 5 to 8 Canvanauts through effective hiring, coaching, succession planning, and retention strategies
Promote a safe and healthy culture with a focus on collaboration and open communication
Set up and run ongoing feedback sessions, initiatives to help drive a healthy code review culture, knowledge sharing, design showcases, and helping improve processes through retrospectives
Required Skills & Background
Experience with technical and people leadership - having previously led small high-performing teams.
Coding proficiency and well as previous experience as a software engineer with a specialization in backend development or data systems
Happy to work in our environment - our core stack is Java and Typescript (some Python and Bash sneaking in here and there) sitting on AWS infrastructure
Capable of building consensus and getting buy-in with multidisciplinary stakeholders
Excellent written and verbal communication skills; with the ability to work with a diverse range of individuals from different backgrounds and with different professional and personal needs
Demonstrable product and commercial acumen
Be capable of leading by example - promoting Canva's values, no-blame mentality, and inclusive culture

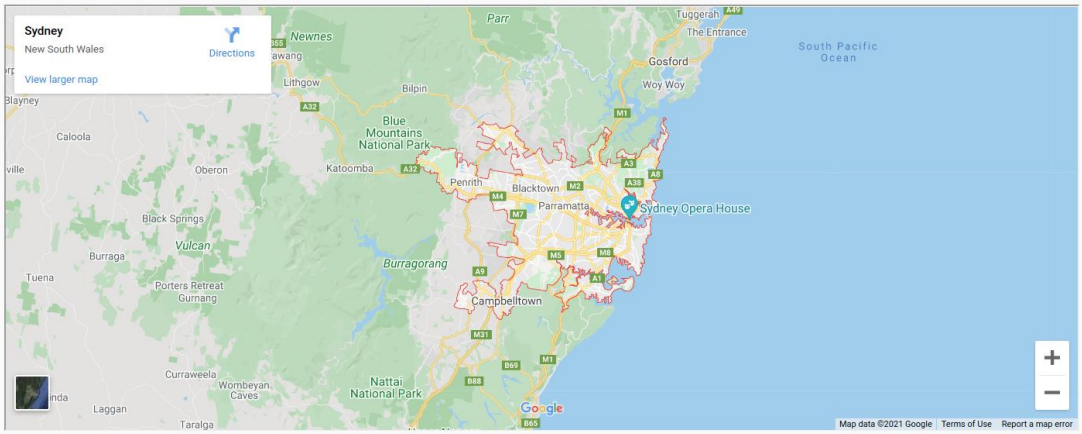
Home
Job Dashboard
Search For Jobs
Statistics
Profile
Settings

- Competitive salary, plus stock options via our ESOP plan
- Flexible daily working hours, we value work-life balance
- Breakfast and lunch prepared by our wonderful Vibe team in Sydney HQ
- Onsite-Gym and Yoga Membership
- End-of-Trip Facilities: Bicycle parking and showers
- Generous parental (including secondary) leave policy
- Pet-friendly offices
- Internal Coaching and Employee Support Programs
- Sponsored social clubs, team events, and celebrations

This role is open to remote applicants within Australia that do not need visa sponsorship. We will require occasional travel to Sydney up to three times per quarter, once it is safe to do so.
The Canva workspace is set up for you to do your best work and have a great time doing so, with breakout areas, little nooks, and quiet places to retreat.
If you're seeking professional growth and enjoy working on complex, large-scale, online products that delight our millions of individual and business users alike all over the world - then apply now to be considered for the position!
We will not accept or review any CVs from external recruitment agencies.

Canva

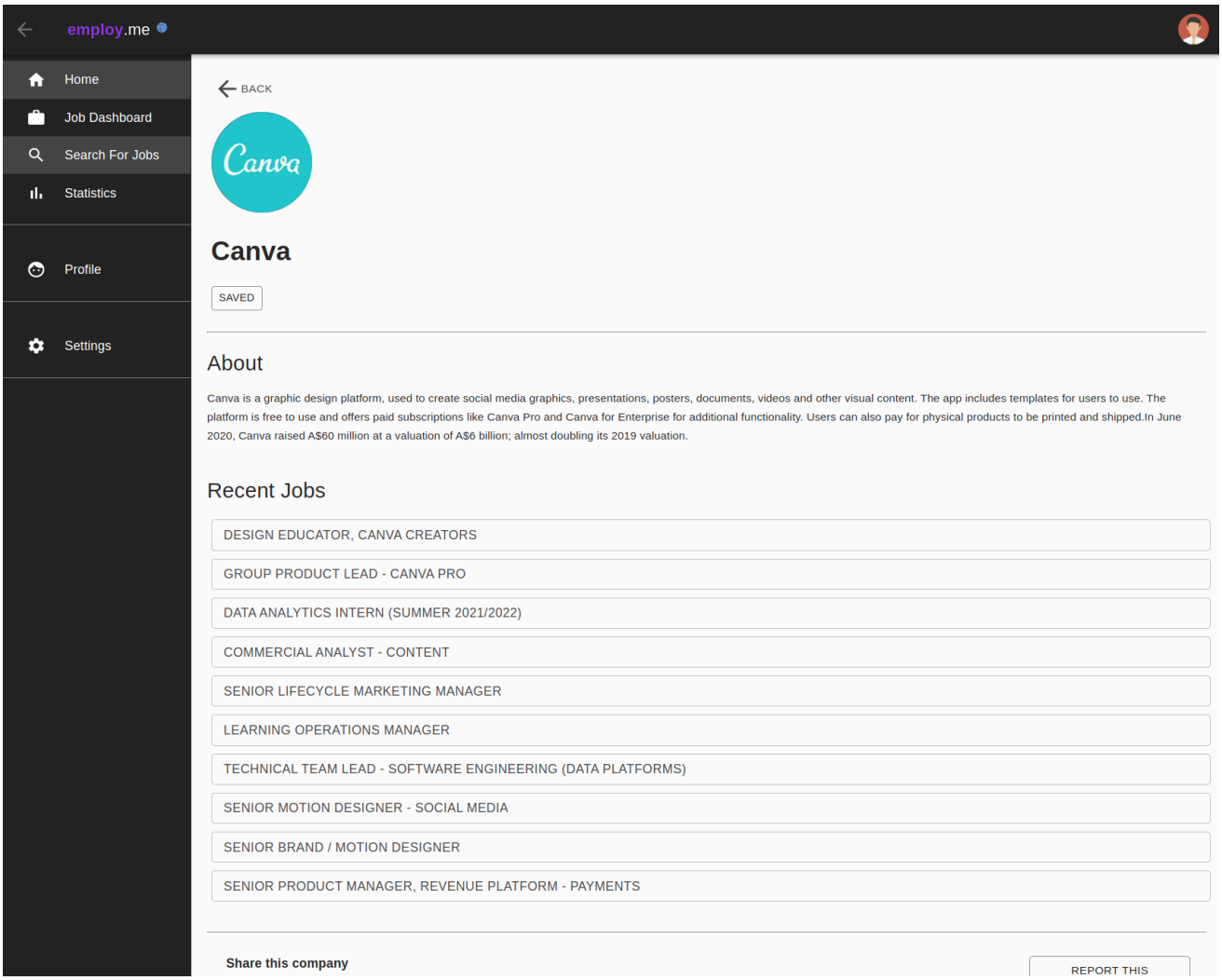
Location



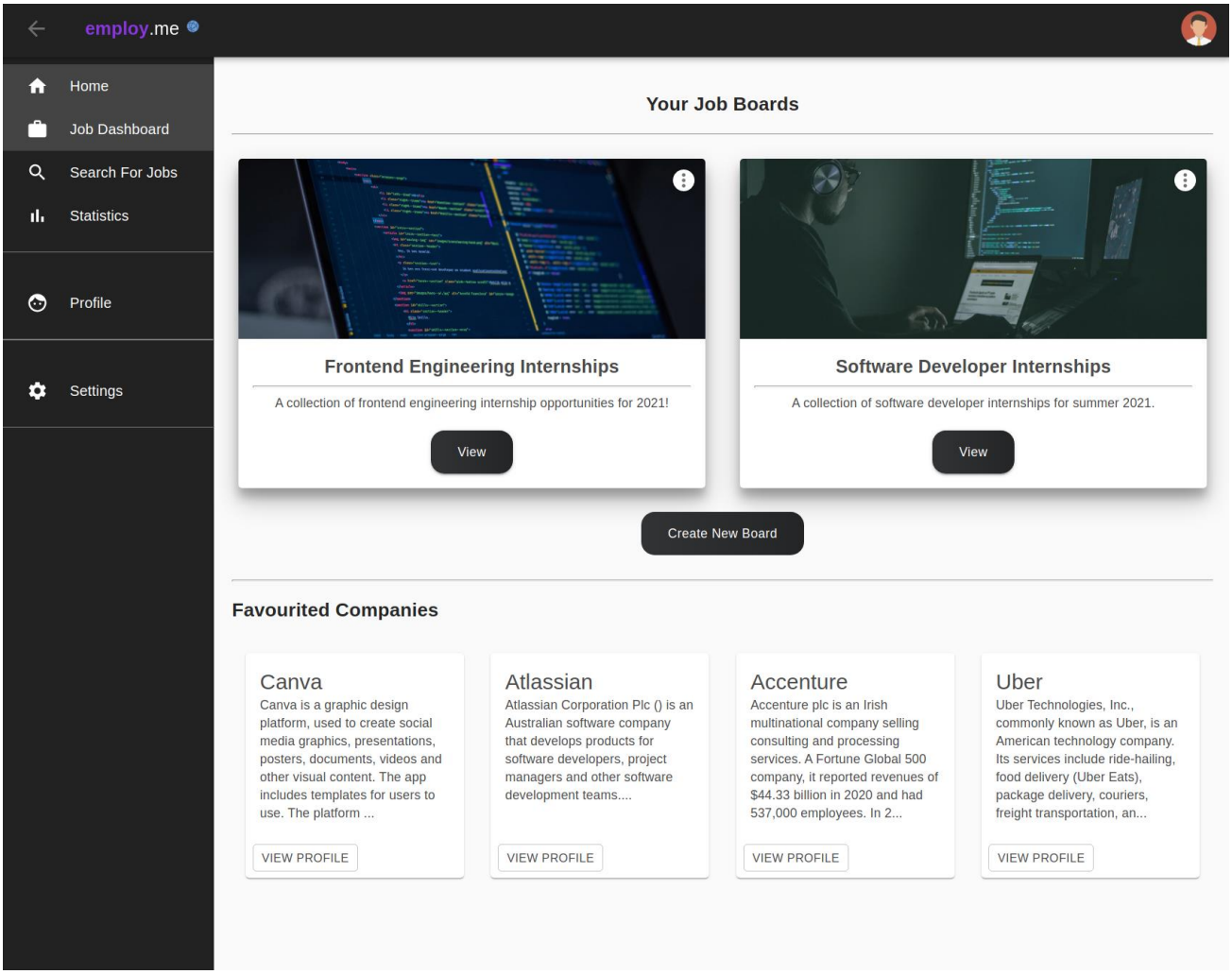
Share this job

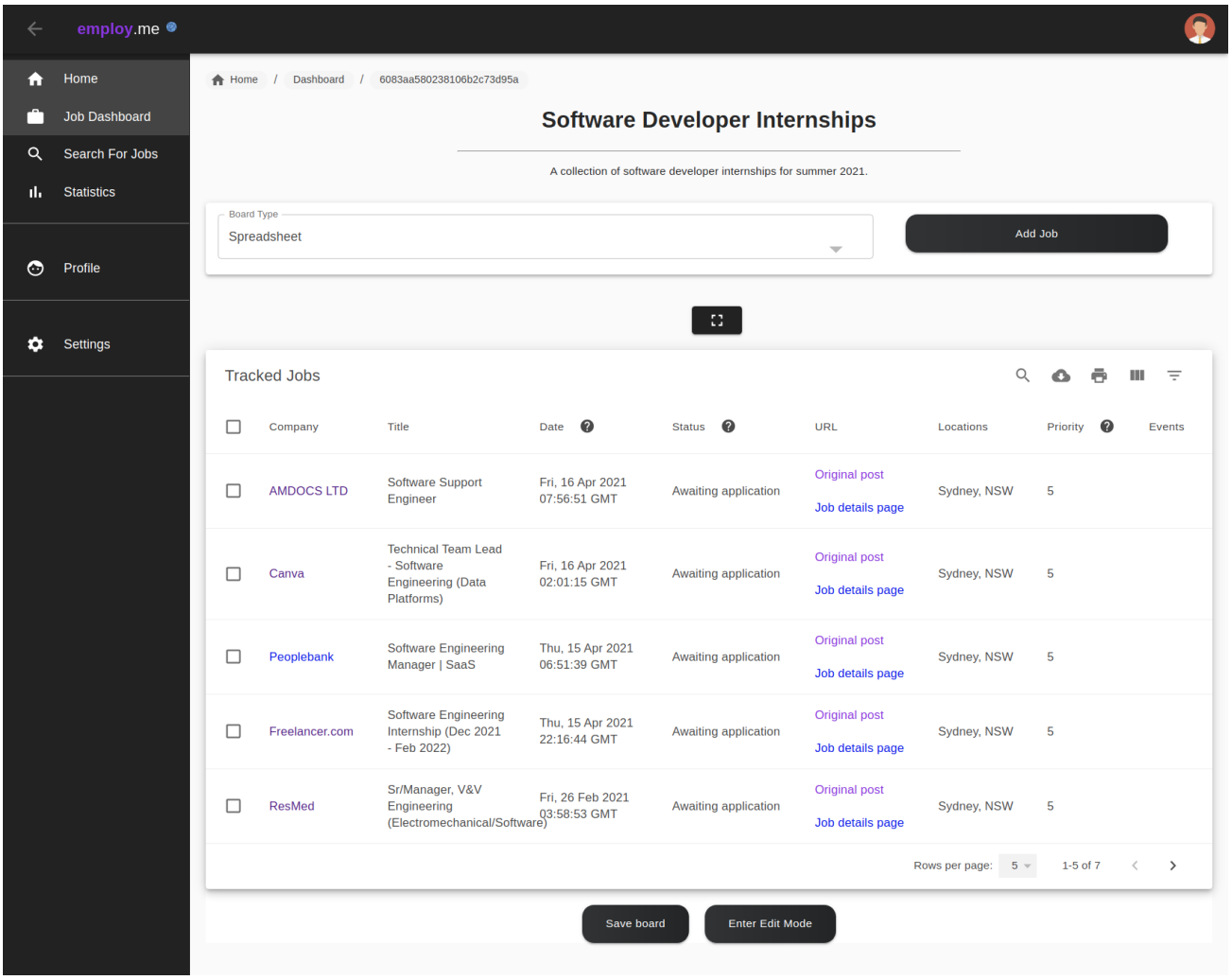
REPORT THIS JOB

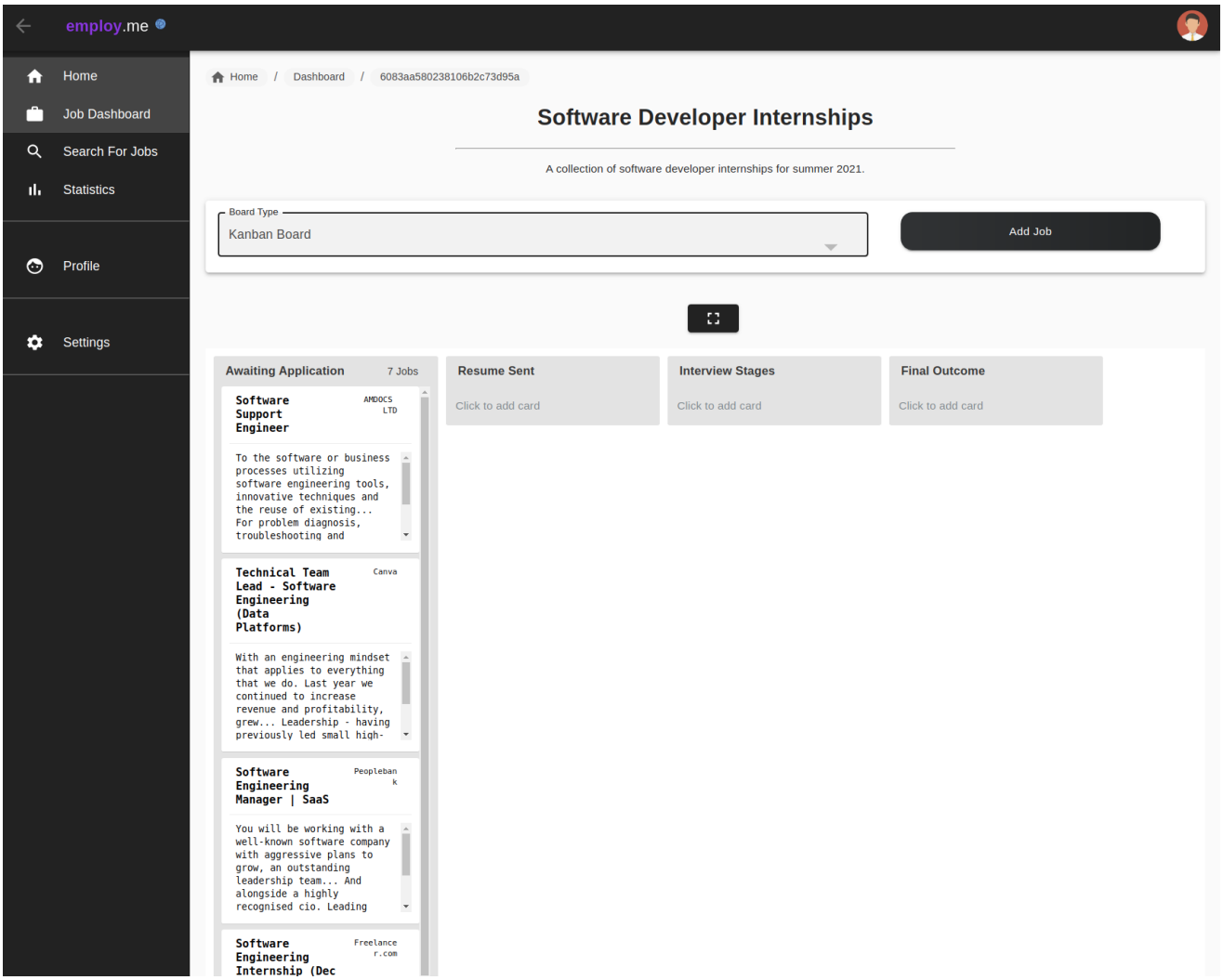
Company Details:

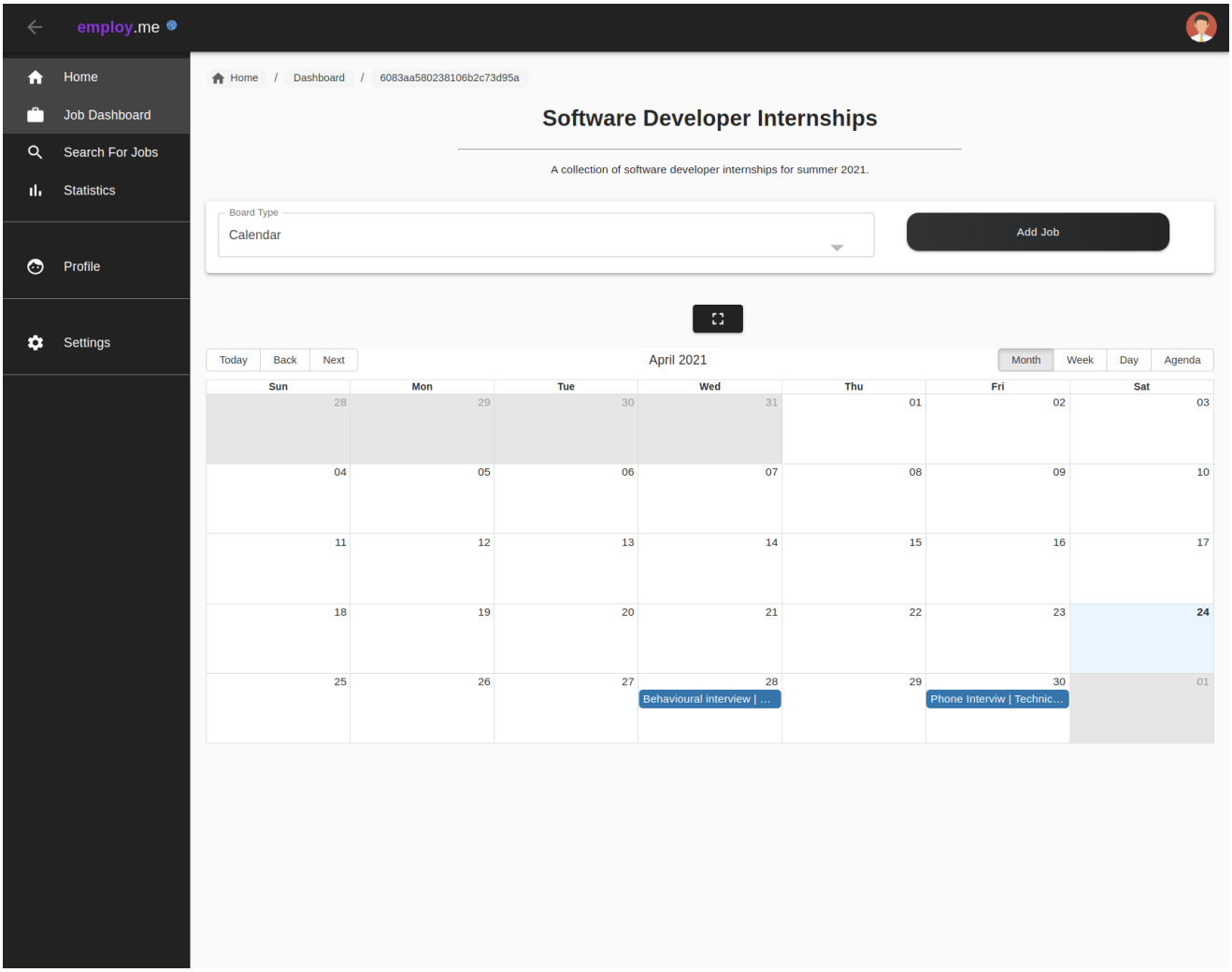


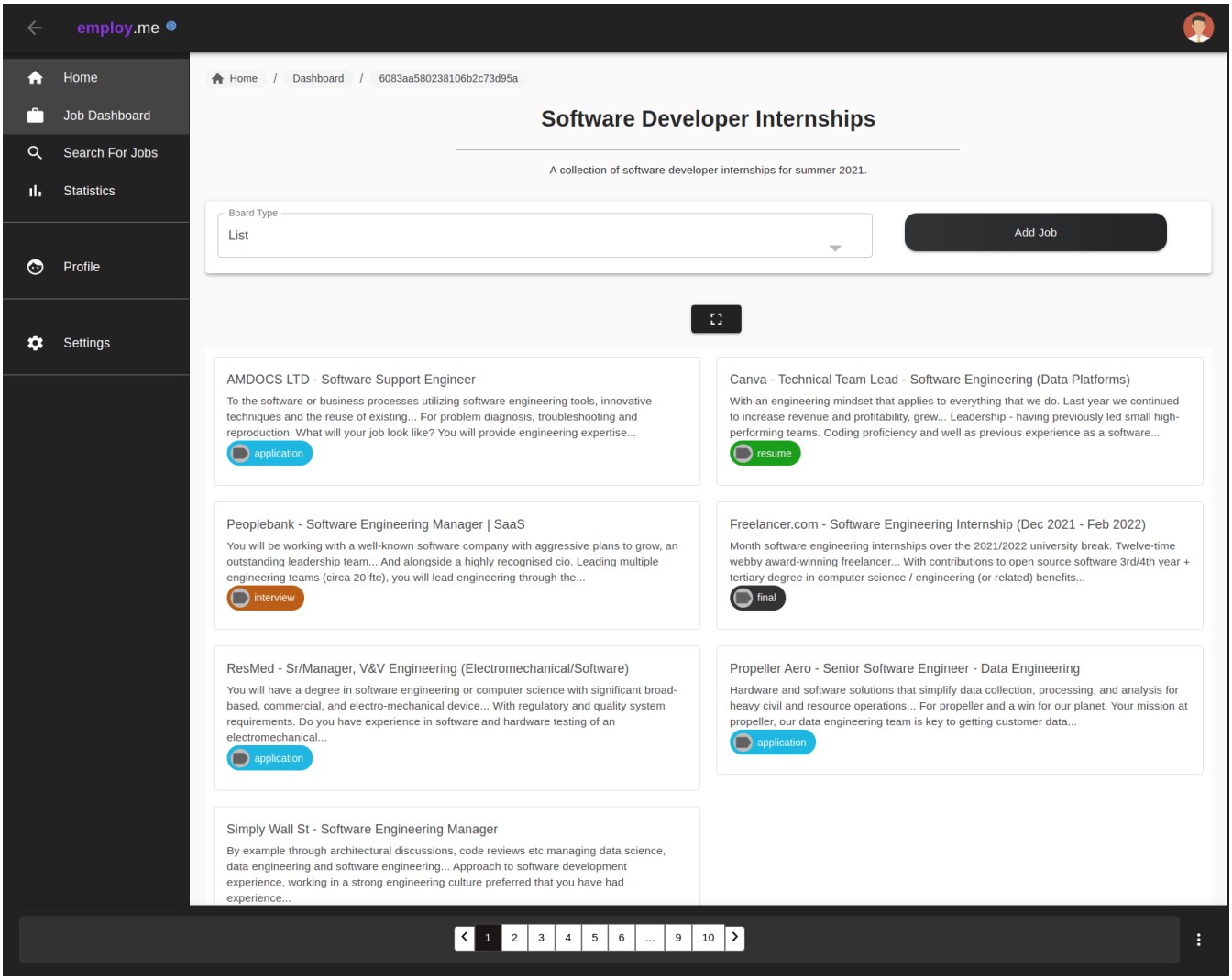
Dashboard:


employ.me
Home
Job Dashboard
Search For Jobs
Statistics
Profile
Settings
Your Job Boards
Frontend Engineering Internships
A collection of frontend engineering internship opportunities for 2021!
View
Software Developer Internships
A collection of software developer internships for summer 2021.
View
Create New Board
Favourited Companies
Canva
Canva is a graphic design platform, used to create social media graphics, presentations, posters, documents, videos and other visual content. The app includes templates for users to use. The platform ...
VIEW PROFILE
Atlassian
Atlassian Corporation Plc () is an Australian software company that develops products for software developers, project managers and other software development teams....
VIEW PROFILE
Accenture
Accenture plc is an Irish multinational company selling consulting and processing services. A Fortune Global 500 company, it reported revenues of $44.33 billion in 2020 and had 537,000 employees. In 2...
VIEW PROFILE
Uber
Uber Technologies, Inc., commonly known as Uber, is an American technology company. Its services include ride-hailing, food delivery (Uber Eats), package delivery, couriers, freight transportation, an...
VIEW PROFILE

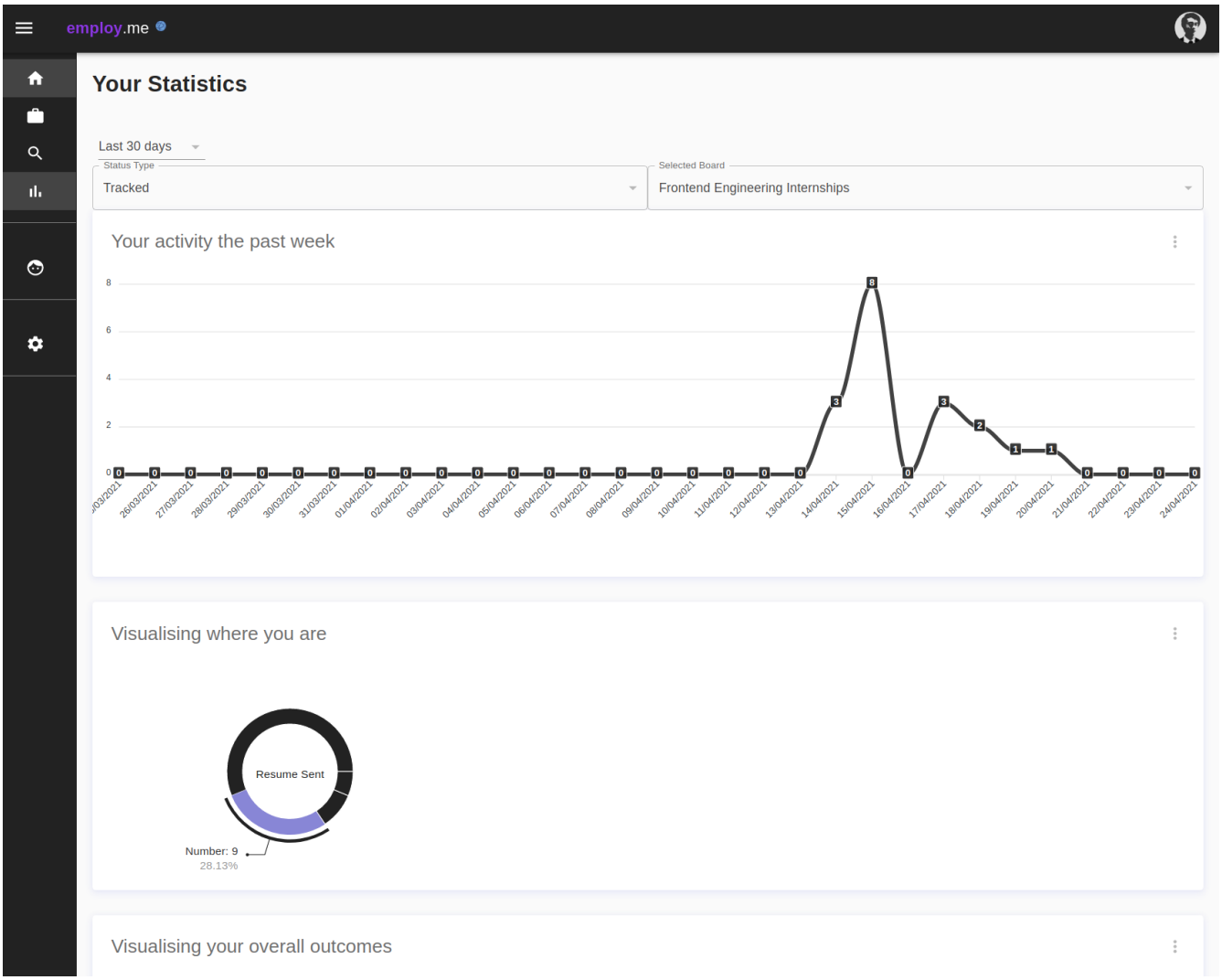

employ.me
Home
Job Dashboard
Search For Jobs
Statistics
Profile
Settings
Home / Dashboard / 6083aa580238106b2c73d95a
Software Developer Internships
A collection of software developer internships for summer 2021.
Board Type
Spreadsheet
Add Job
Tracked Jobs
Company
Title
Date
Status
URL
Locations
Priority
Events
AMDOCS LTD
Software Support Engineer
Fri, 16 Apr 2021 07:56:51 GMT
Awaiting application
Original post
Job details page
Sydney, NSW
5
Canva
Technical Team Lead - Software Engineering (Data Platforms)
Fri, 16 Apr 2021 02:01:15 GMT
Awaiting application
Original post
Job details page
Sydney, NSW
5
Peoplebank
Software Engineering Manager | SaaS
Thu, 15 Apr 2021 06:51:39 GMT
Awaiting application
Original post
Job details page
Sydney, NSW
5
Freelancer.com
Software Engineering Internship (Dec 2021 - Feb 2022)
Thu, 15 Apr 2021 22:16:44 GMT
Awaiting application
Original post
Job details page
Sydney, NSW
5
ResMed
Sr/Manager, V&V Engineering (Electromechanical/Software)
Fri, 26 Feb 2021 03:58:53 GMT
Awaiting application
Original post
Job details page
Sydney, NSW
5
Rows per page: 5
1-5 of 7
Save board
Enter Edit Mode

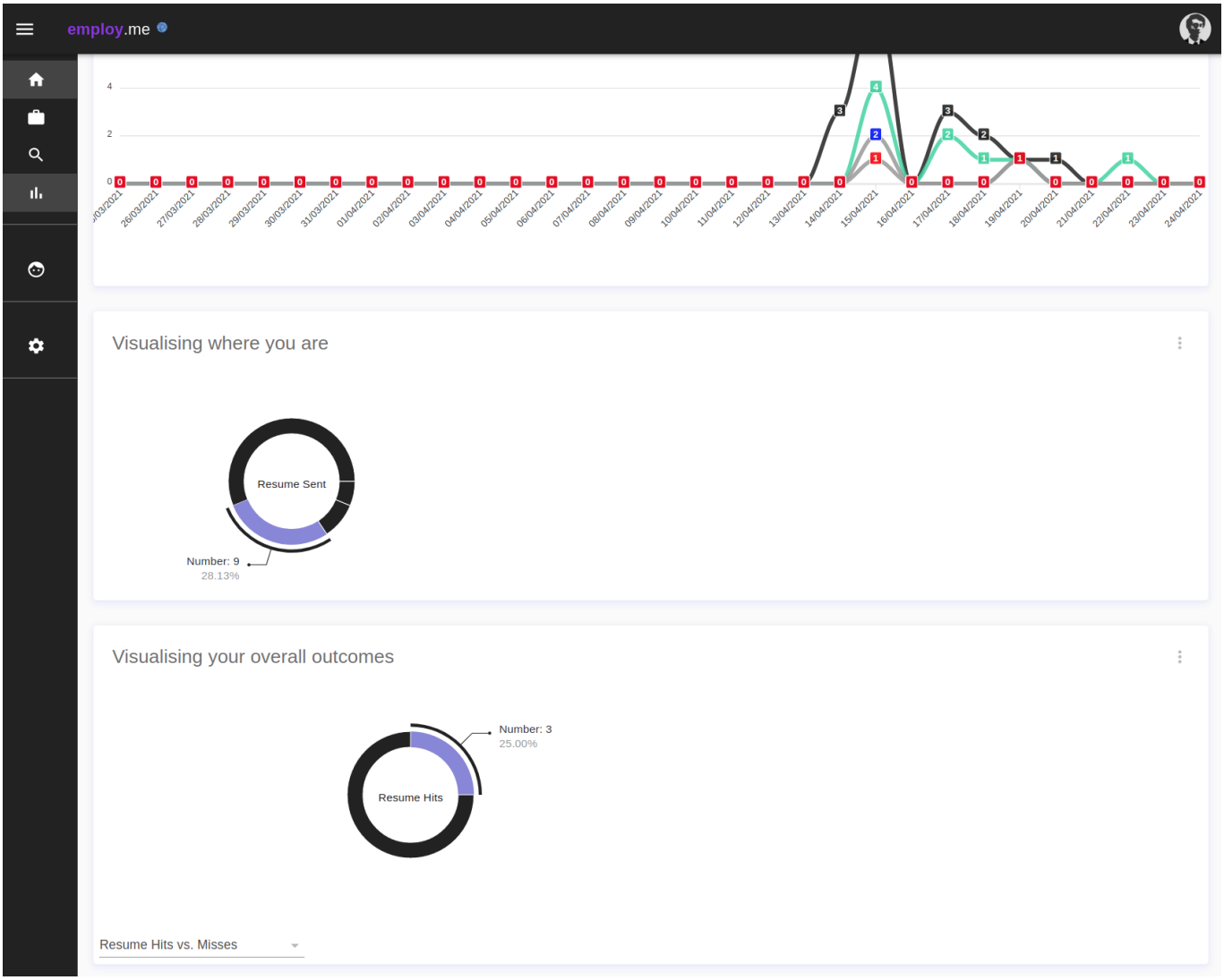

employ.me
Home
Job Dashboard
Search For Jobs
Statistics
Profile
Settings
Home / Dashboard / 6083aa580238106b2c73d95a
Software Developer Internships
A collection of software developer internships for summer 2021.
Board Type
Kanban Board
Add Job
Awaiting Application
7 Jobs
Software Support Engineer
AMDOCS LTD
To the software or business processes utilizing software engineering tools, innovative techniques and the reuse of existing... For problem diagnosis, troubleshooting and
Technical Team Lead - Software Engineering (Data Platforms)
Canva
With an engineering mindset that applies to everything that we do. Last year we continued to increase revenue and profitability, grew... Leadership - having previously led small high-
Software Engineering Manager | SaaS
Peopleban k
You will be working with a well-known software company with aggressive plans to grow, an outstanding leadership team... And alongside a highly recognised cio. Leading
Software Engineering Internship (Dec
Freelance r.com
Resume Sent
Click to add card
Interview Stages
Click to add card
Final Outcome
Click to add card


employ.me
Home
Job Dashboard
Search For Jobs
Statistics
Profile
Settings
Home / Dashboard / 6083aa580238106b2c73d95a
Software Developer Internships
A collection of software developer internships for summer 2021.
Board Type
Calendar
Add Job
Today
Back
Next
April 2021
Month
Week
Day
Agenda
Sun
Mon
Tue
Wed
Thu
Fri
Sat
28
29
30
31
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
01
Behavioural interview | ...
Phone Interviw | Technic...

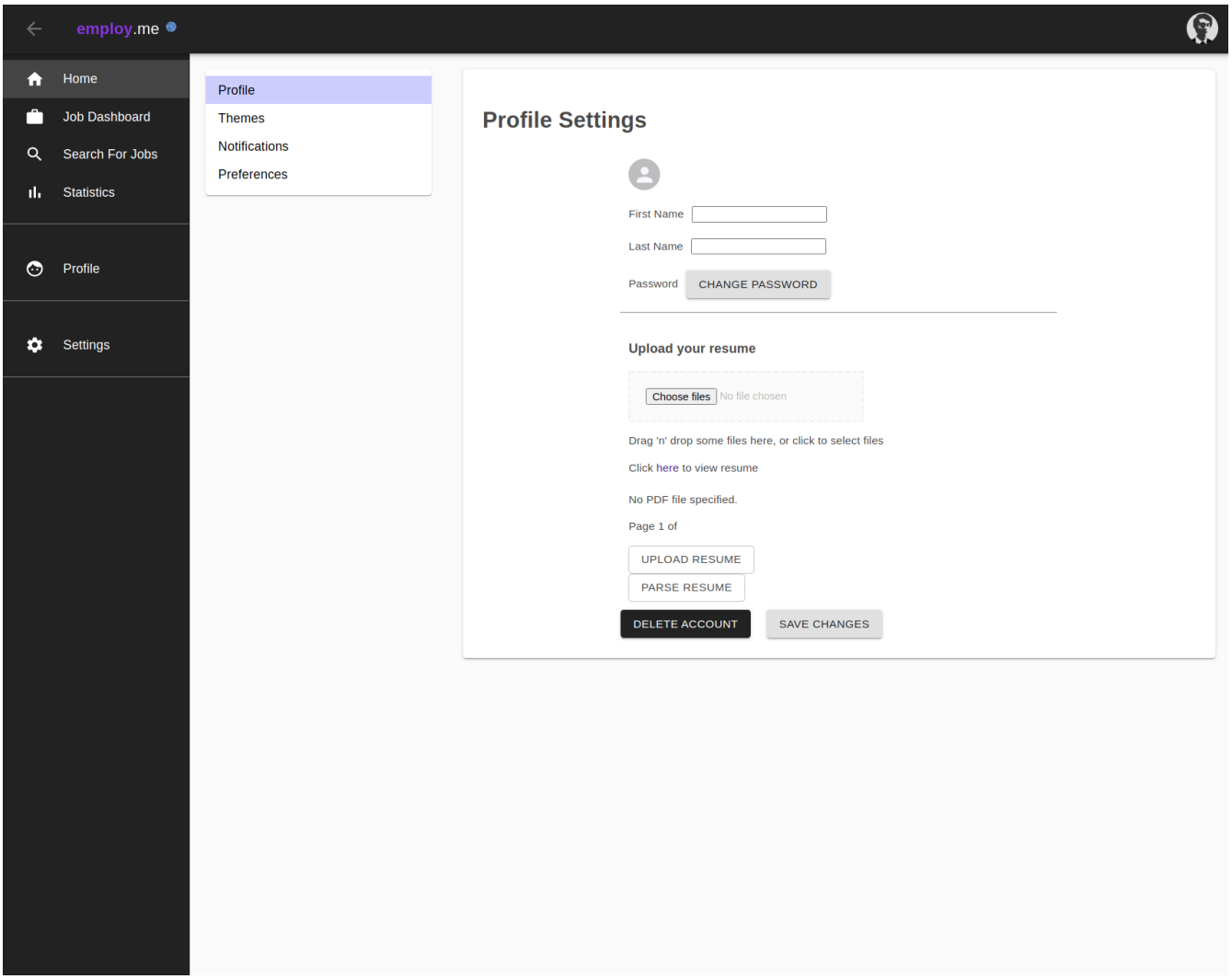

employ.me
Home
Job Dashboard
Search For Jobs
Statistics
Profile
Settings
Home / Dashboard / 6083aa580238106b2c73d95a
Software Developer Internships
A collection of software developer internships for summer 2021.
Board Type
List
Add Job
AMDOCS LTD - Software Support Engineer
To the software or business processes utilizing software engineering tools, innovative techniques and the reuse of existing... For problem diagnosis, troubleshooting and reproduction. What will your job look like? You will provide engineering expertise...
application
Canva - Technical Team Lead - Software Engineering (Data Platforms)
With an engineering mindset that applies to everything that we do. Last year we continued to increase revenue and profitability, grew... Leadership - having previously led small high-performing teams. Coding proficiency and well as previous experience as a software...
resume
Peoplebank - Software Engineering Manager | SaaS
You will be working with a well-known software company with aggressive plans to grow, an outstanding leadership team... And alongside a highly recognised cio. Leading multiple engineering teams (circa 20 fte), you will lead engineering through the...
interview
Freelancer.com - Software Engineering Internship (Dec 2021 - Feb 2022)
Month software engineering internships over the 2021/2022 university break. Twelve-time webby award-winning freelancer... With contributions to open source software 3rd/4th year + tertiary degree in computer science / engineering (or related) benefits...
final
ResMed - Sr/Manager, V&V Engineering (Electromechanical/Software)
You will have a degree in software engineering or computer science with significant broad-based, commercial, and electro-mechanical device... With regulatory and quality system requirements. Do you have experience in software and hardware testing of an electromechanical...
application
Propeller Aero - Senior Software Engineer - Data Engineering
Hardware and software solutions that simplify data collection, processing, and analysis for heavy civil and resource operations... For propeller and a win for our planet. Your mission at propeller, our data engineering team is key to getting customer data...
application
Simply Wall St - Software Engineering Manager
By example through architectural discussions, code reviews etc managing data science, data engineering and software engineering... Approach to software development experience, working in a strong engineering culture preferred that you have had experience...
1 2 3 4 5 6 ... 9 10

Statistics:

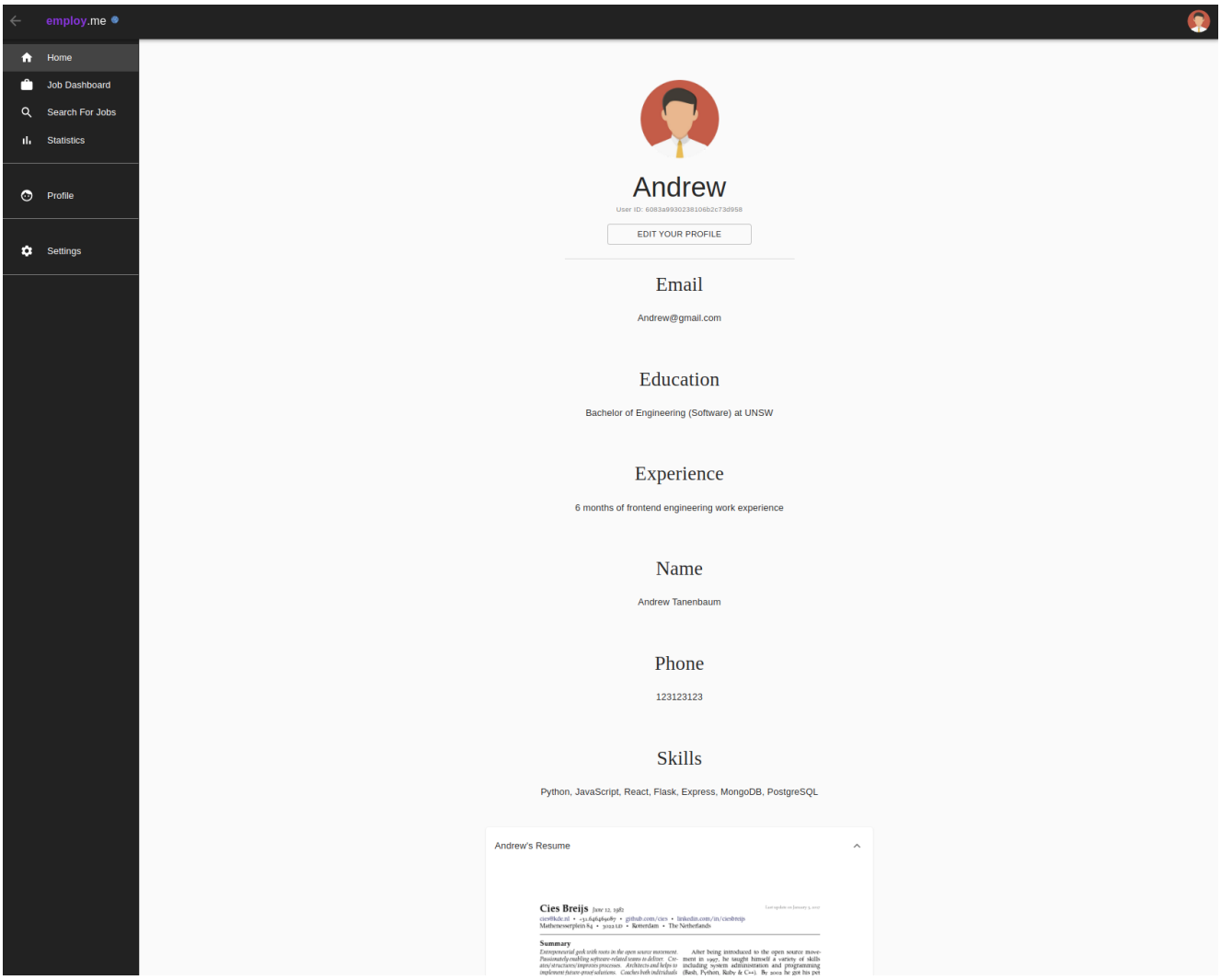

employ.me
Your Statistics
Last 30 days
Status Type
Tracked
Selected Board
Frontend Engineering Internships
Your activity the past week
Visualising where you are
Resume Sent
Number: 9
28.13%
Visualising your overall outcomes

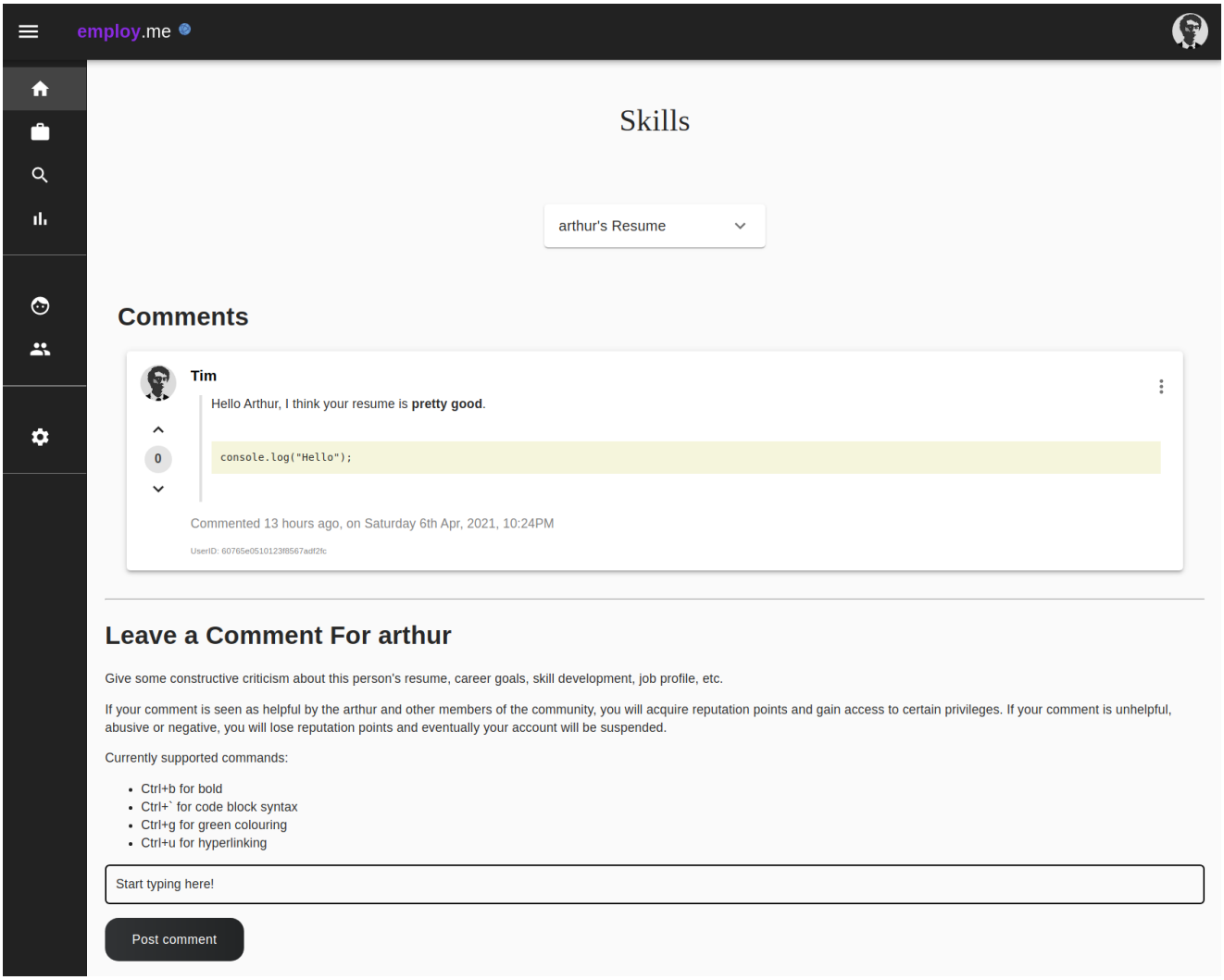

employ.me
4
2
0
Visualising where you are
Resume Sent
Number: 9
28.13%
Visualising your overall outcomes
Number: 3
25.00%
Resume Hits
Resume Hits vs. Misses

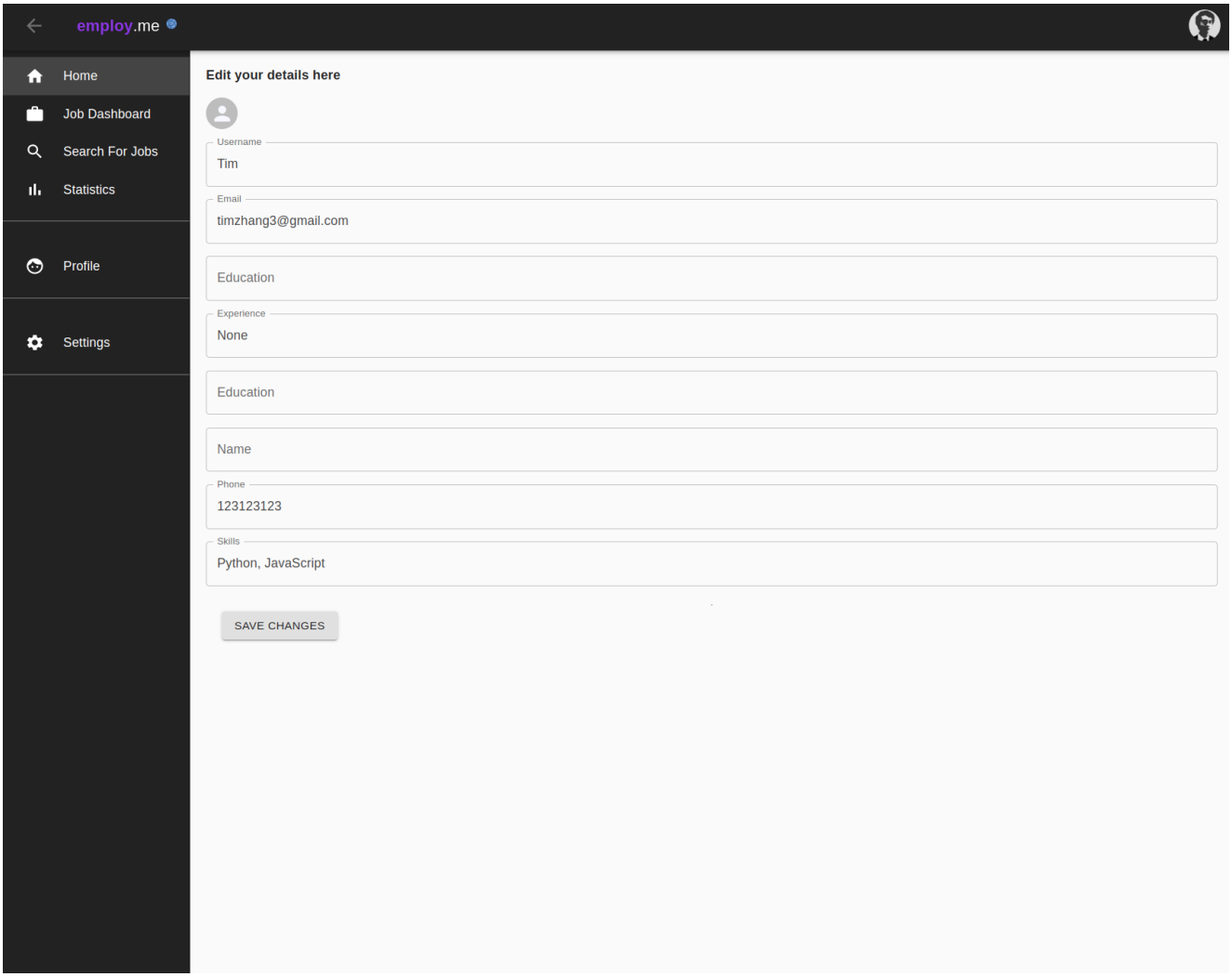
Settings:


employ.me
Home
Job Dashboard
Search For Jobs
Statistics
Profile
Settings
Profile
Themes
Notifications
Preferences
Profile Settings
First Name
Last Name
Password
CHANGE PASSWORD
Upload your resume
Choose files
No file chosen
Drag 'n' drop some files here, or click to select files
Click here to view resume
No PDF file specified.
Page 1 of
UPLOAD RESUME
PARSE RESUME
DELETE ACCOUNT
SAVE CHANGES

Profile:


employ.me
Home
Job Dashboard
Search For Jobs
Statistics
Profile
Settings
Andrew
User ID: 6083a993023810 6b2c73d958
EDIT YOUR PROFILE
Email
Andrew@gmail.com
Education
Bachelor of Engineering (Software) at UNSW
Experience
6 months of frontend engineering work experience
Name
Andrew Tanenbaum
Phone
123123123
Skills
Python, JavaScript, React, Flask, Express, MongoDB, PostgreSQL
Andrew's Resume
Cies Breijs
Summary


employ.me
Skills
arthur's Resume
Comments
Tim
Hello Arthur, I think your resume is pretty good.
console.log("Hello");
0
Commented 13 hours ago, on Saturday 6th Apr, 2021, 10:24PM
UserID: 60765e0510123f8567adf2fc
Leave a Comment For arthur
Give some constructive criticism about this person's resume, career goals, skill development, job profile, etc.
If your comment is seen as helpful by the arthur and other members of the community, you will acquire reputation points and gain access to certain privileges. If your comment is unhelpful, abusive or negative, you will lose reputation points and eventually your account will be suspended.
Currently supported commands:
Ctrl+b for bold
Ctrl+` for code block syntax
Ctrl+g for green colouring
Ctrl+u for hyperlinking
Start typing here!
Post comment


employ.me
Home
Job Dashboard
Search For Jobs
Statistics
Profile
Settings
Edit your details here
Username
Tim
Email
timzhang3@gmail.com
Education
Experience
None
Education
Name
Phone
123123123
Skills
Python, JavaScript
SAVE CHANGES

Part II: Design

Updated Design:

The sections of Part II illustrate the project's architecture, sequence diagrams, system design and technology stack. The following lists describe features that were fully implemented, partially implemented and not implemented.

Fully implemented: the team was able to fully implement all the desired features of the following modules:

- Board automation and job tracking
    - Board creation, editing and deletion of multiple boards owned by a user
    - Editable spreadsheet with utilities such as searching, filtering, sorting, downloadable CSV and printing
    - Usable Kanban board
    - Events creation and rendering on the calendar view
    - Any job post found on the site can be tracked, which will automatically create an entry in a job board
    - External jobs can be manually added for tracking
- Job search
    - Job postings can be queried by title and location
    - Job postings can be sorted and the level of detail can be toggled
- Job details
    - Details regarding a job posting can be fetched and rendered
    - Location of the job is rendered on an embedded Google Maps interface
- Company page
    - Details regarding a specific company and its released jobs can be fetched and rendered dynamically
- User analytics
    - User activity and success can be visualised through interactive line charts and pie charts

Partially implemented: the team implemented a subset of the features for the following modules:

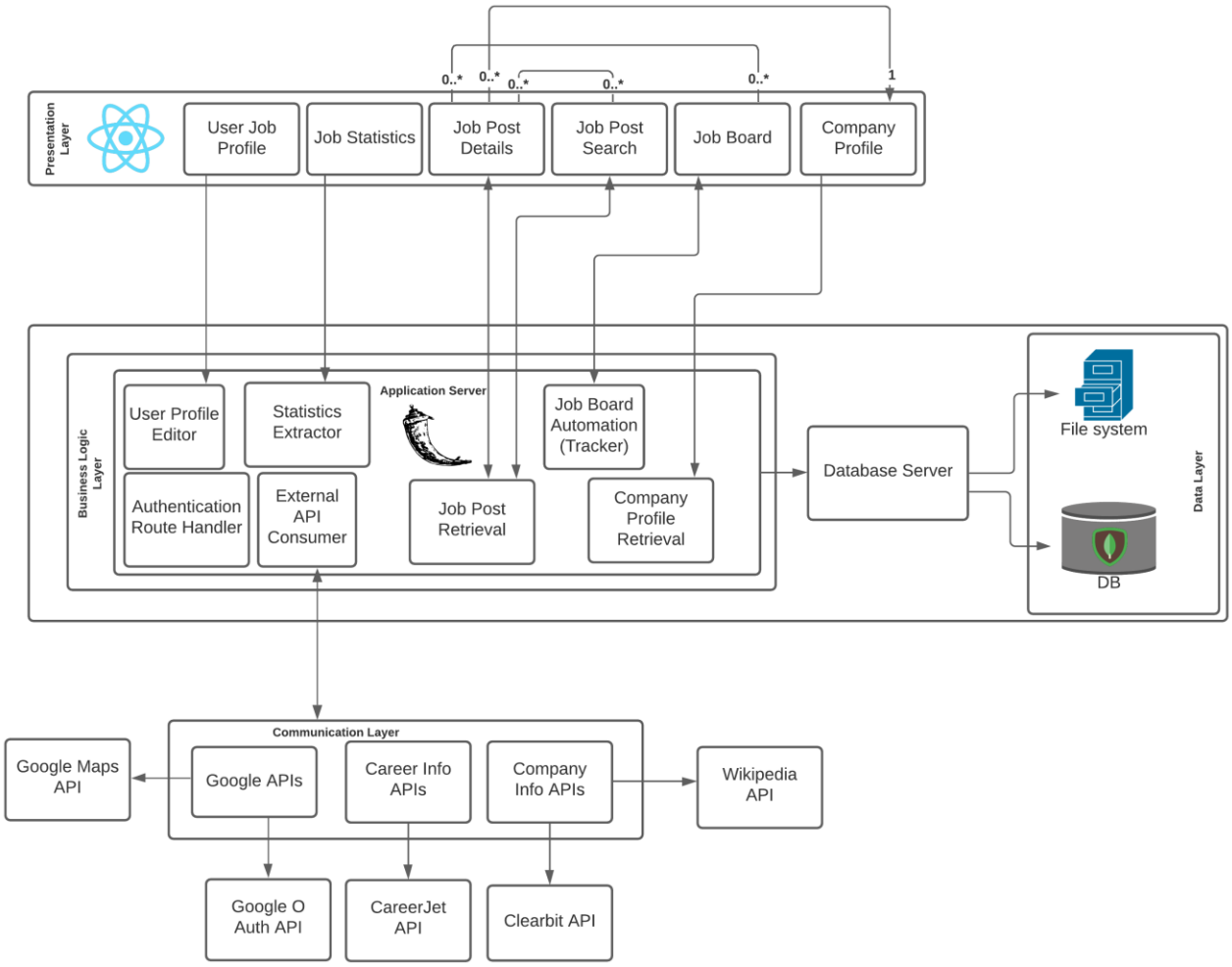
- Community system
    - Users can view each other's public profiles
    - Comments can be posted to any user's profile
    - Rich text editor for comments
- Job recommendation system
    - Resume PDFs can be uploaded by the user and parsed by optical character and text processing models

Not implemented: the team decided against implementing the following feature set:

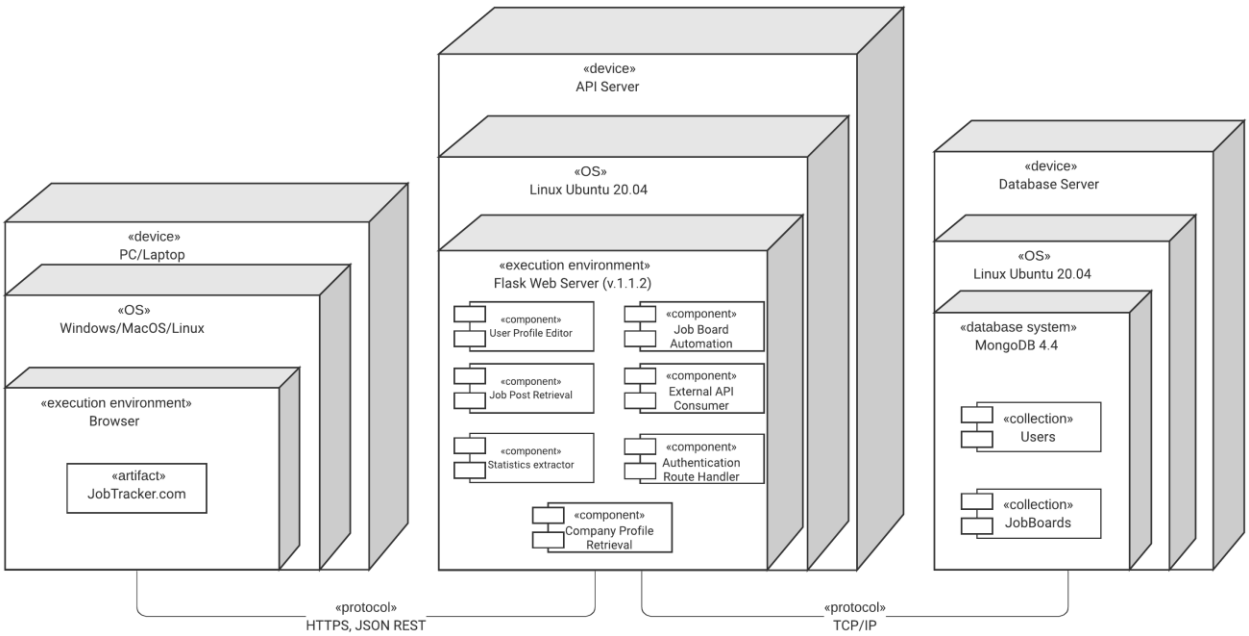
- Job recommendation system
    - Displaying a percentage match between a person's job profile and each job post they find on the application
- Automated notification services
    - Configurable email and SMS notifications for tracked job postings that have an event coming soon
- Community system

- Reputation system
- Searching for specific users
- Public and private profile toggling

Software Architecture:

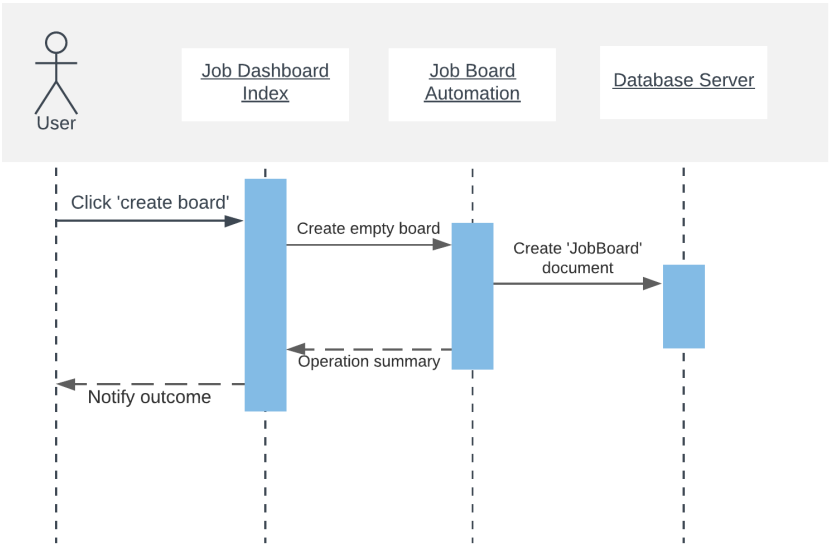


Deployment Diagram

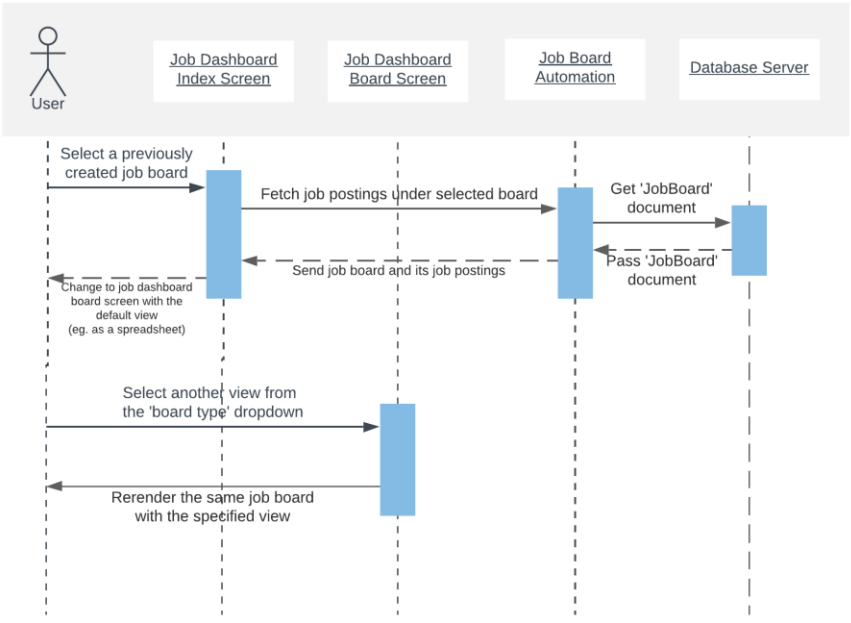


Sequence Diagrams:

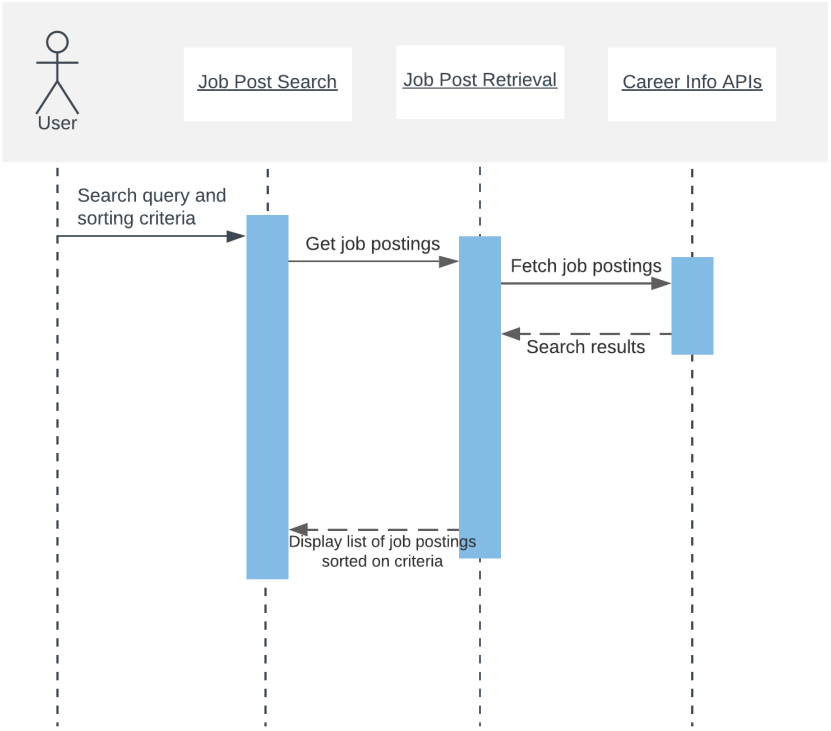
Board Management



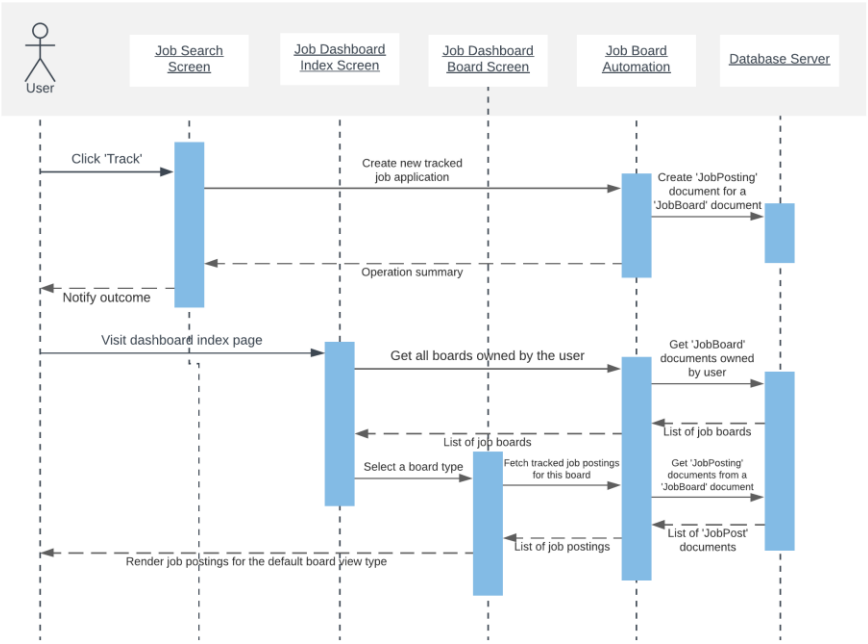
Toggleable Board Views



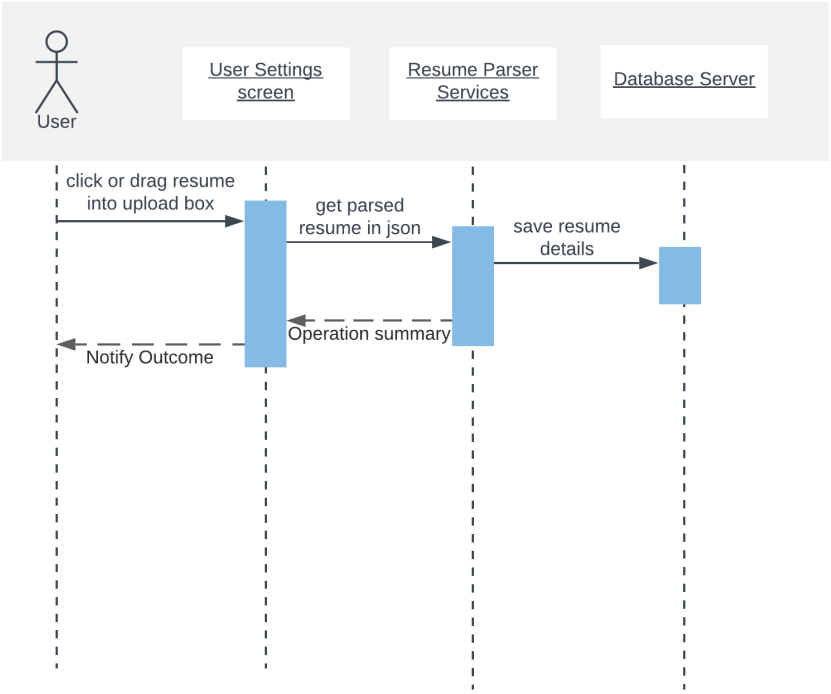
Job Searching and Sorting



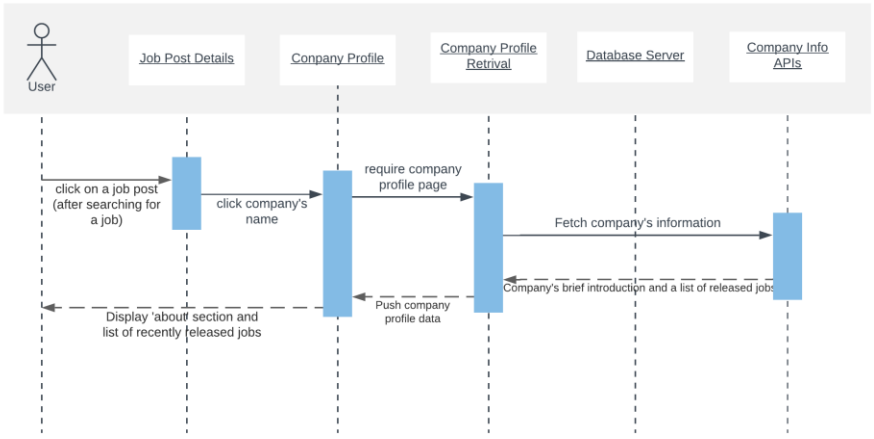
Automated Job Tracking



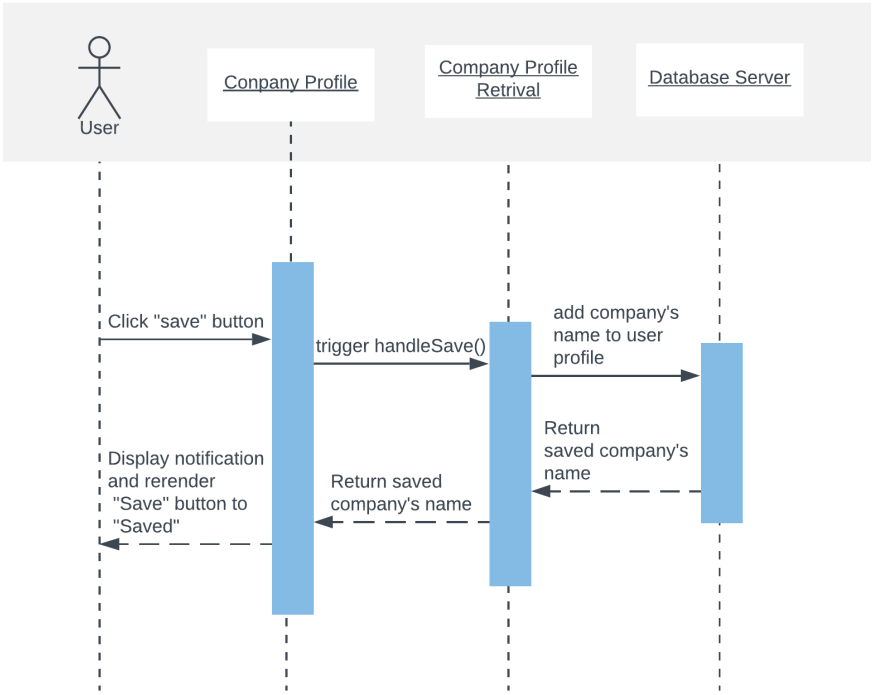
Uploading Resume



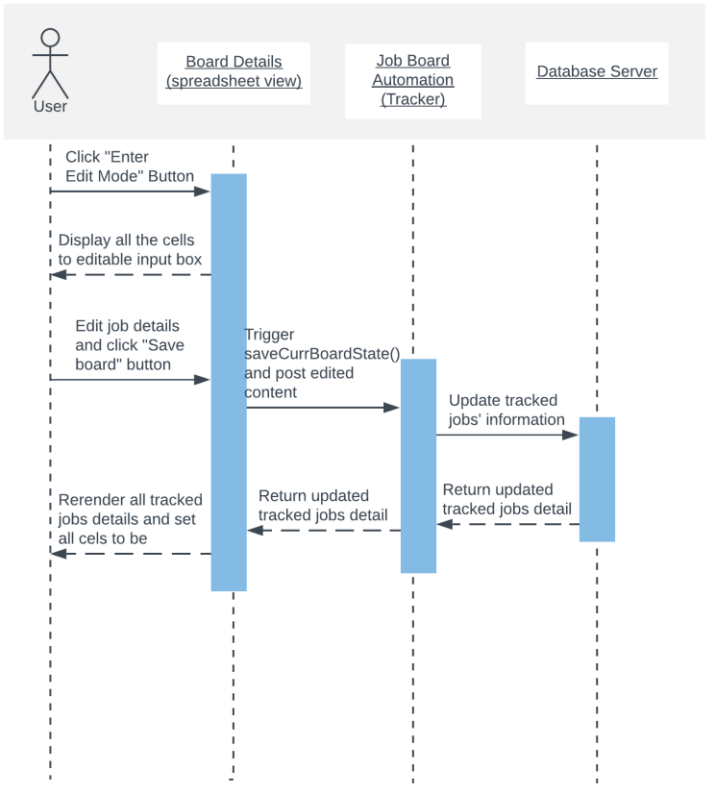
Fetching Company Details



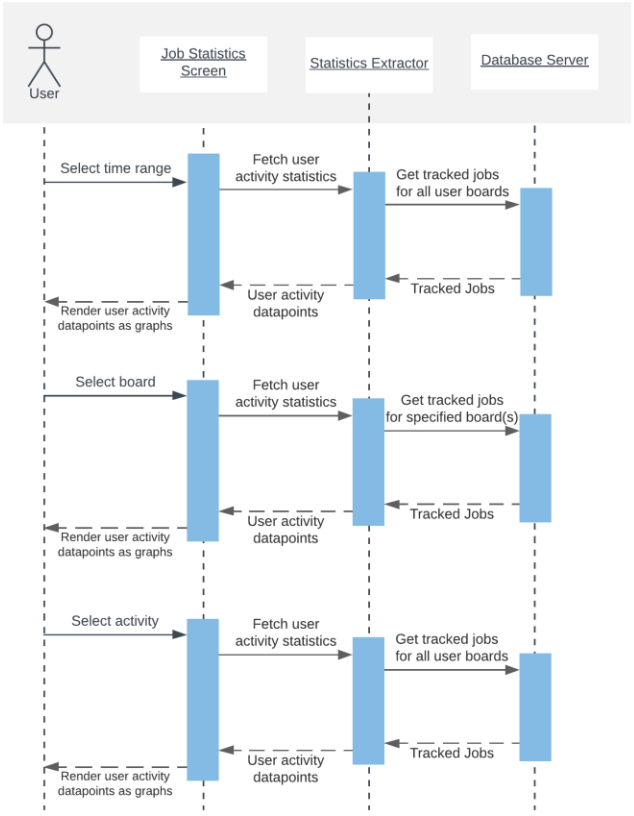
Saving favourite companies



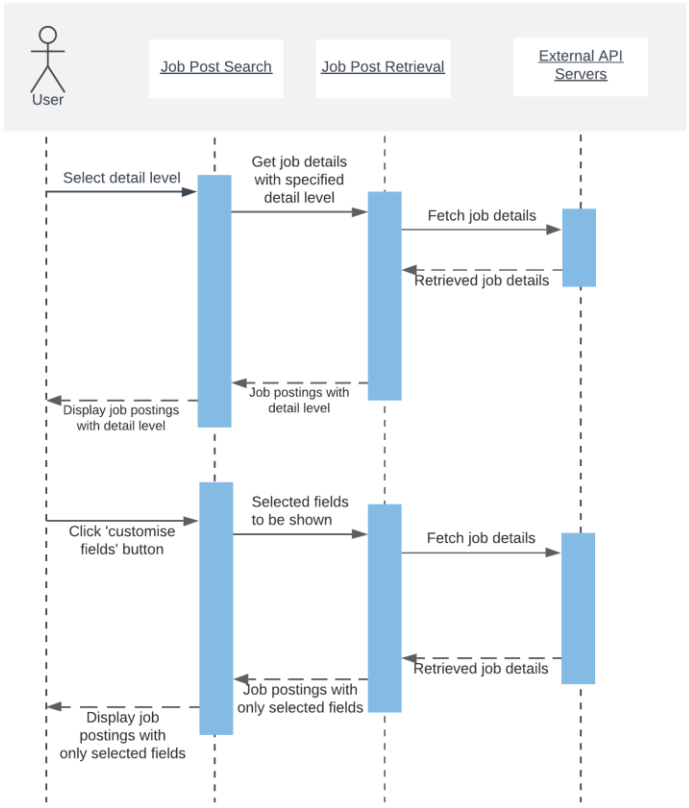
Board Edition



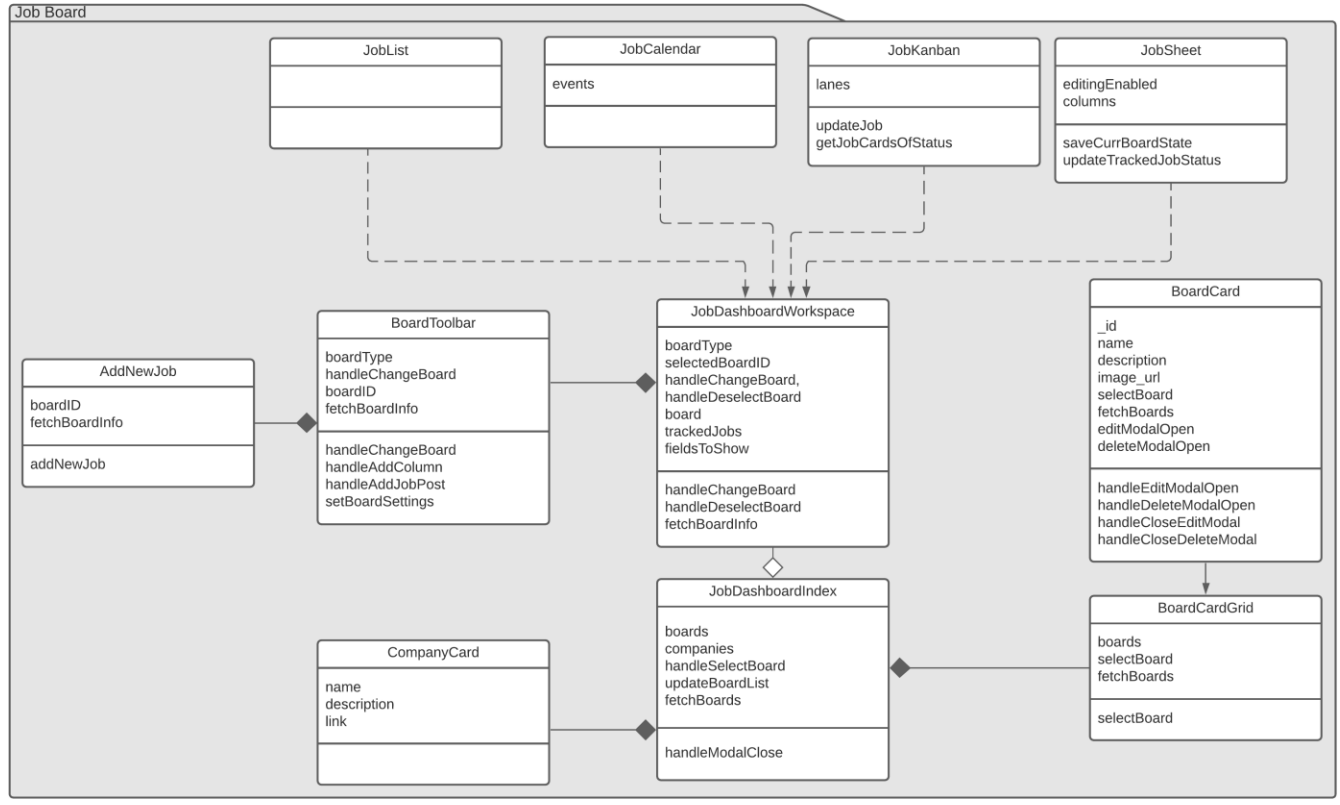
User Analytics Visualisation

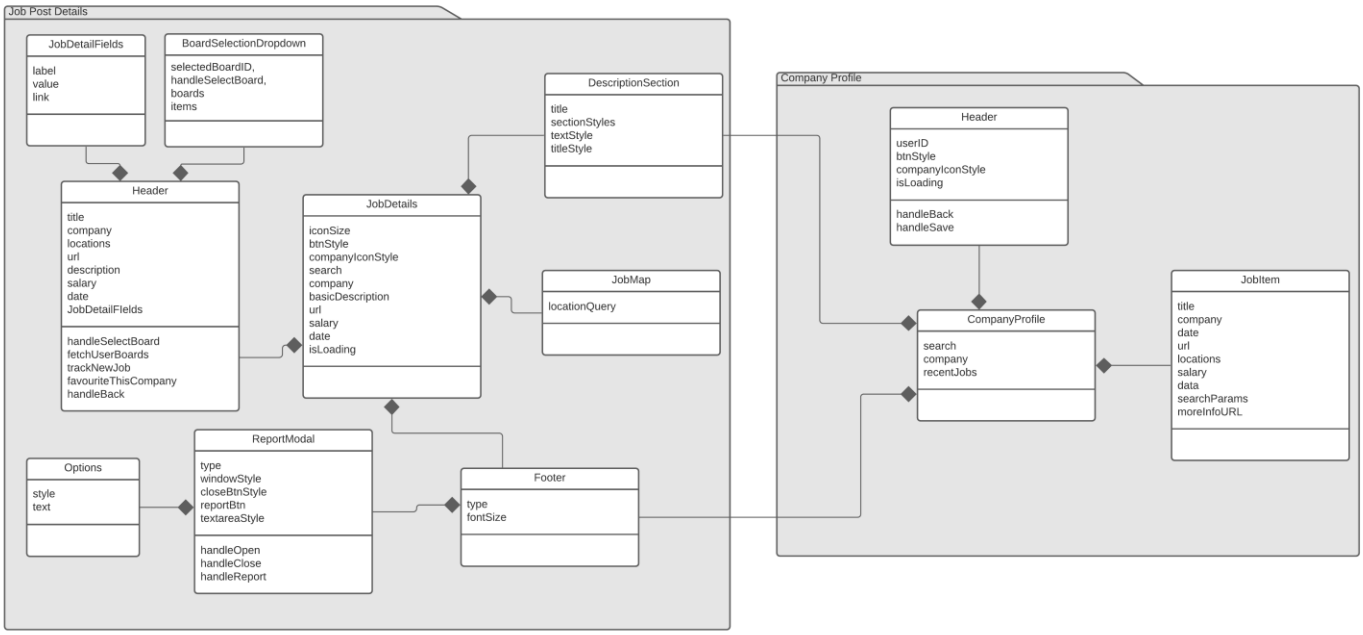


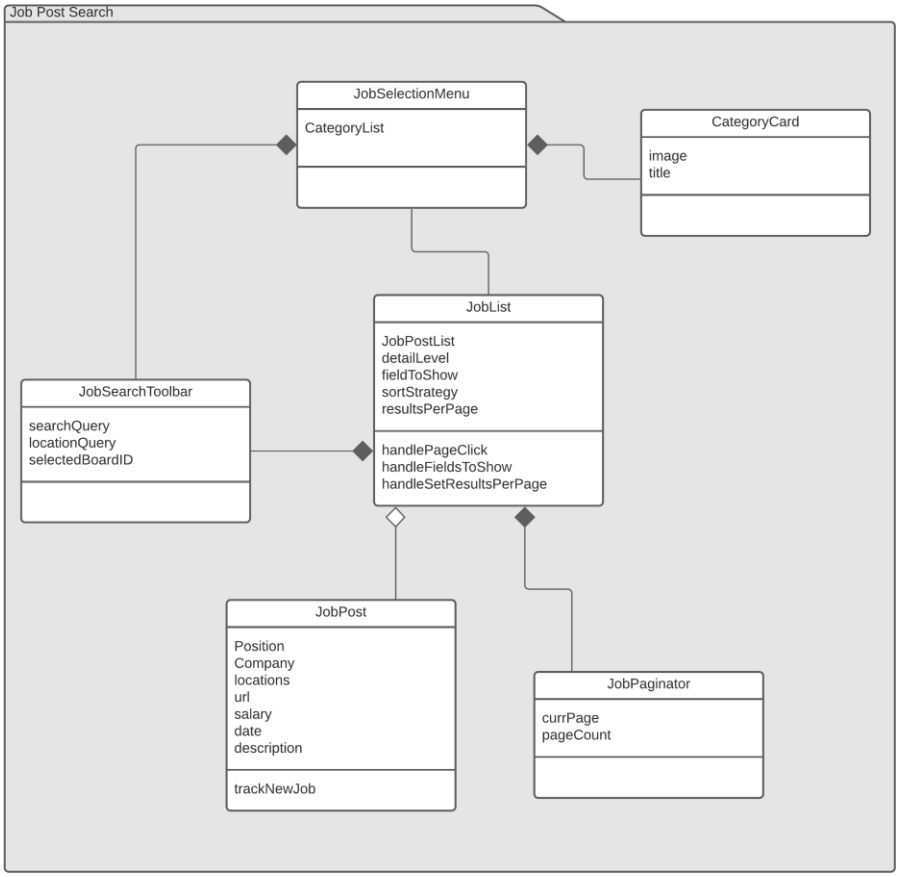
Toggleable level of detail

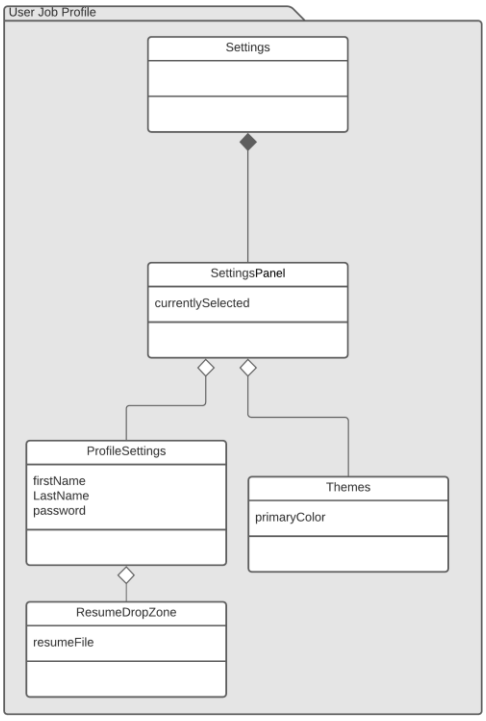


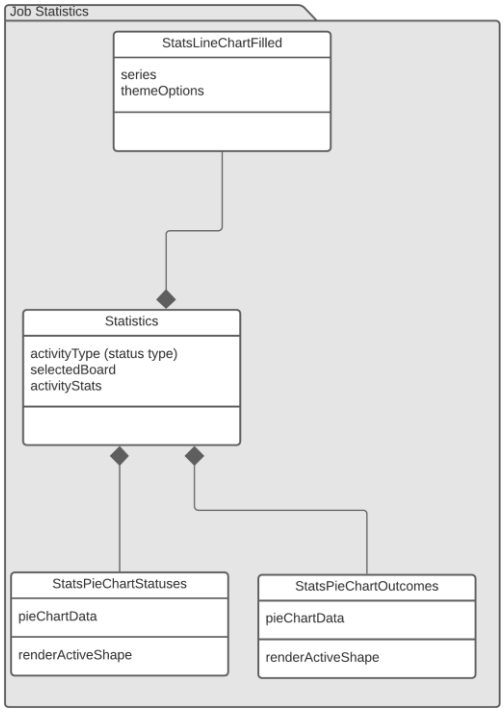
Front-End UML Diagram:



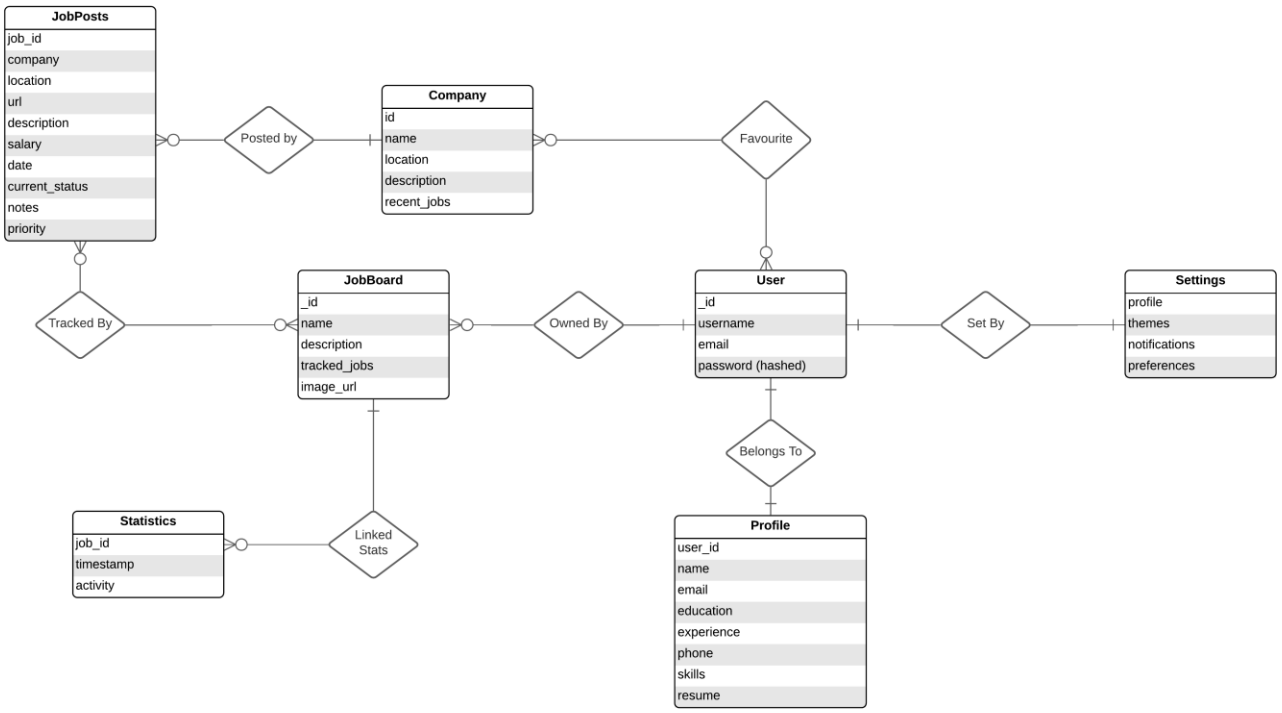



Job Post Search
JobSelectionMenu
CategoryList
CategoryCard
image
title
JobList
JobPostList
detailLevel
fieldToShow
sortStrategy
resultsPerPage
handlePageClick
handleFieldsToShow
handleSetResultsPerPage
JobSearchToolbar
searchQuery
locationQuery
selectedBoardID
JobPost
Position
Company
locations
url
salary
date
description
trackNewJob
JobPaginator
currPage
pageCount


User Job Profile
Settings
SettingsPanel
currentlySelected
ProfileSettings
firstName
LastName
password
Themes
primaryColor
ResumeDropZone
resumeFile


Job Statistics
StatsLineChartFilled
series
themeOptions
Statistics
activityType (status type)
selectedBoard
activityStats
StatsPieChartStatuses
pieChartData
renderActiveShape
StatsPieChartOutcomes
pieChartData
renderActiveShape

Back-End ER Diagram:



Key Technologies:

Front-end:

- Language: JavaScript
- Frontend Framework/Library: React
- UI component library: Material UI

Back-end:

- Database: MongoDB
- API Management Tool: Rest API
- Web Server: Flask with object document mapper (ODM)

APIs:

We used a range of APIs to gather data about companies, jobs and location for our application. Company Information APIs:

1. Clearbit Logo
2. Wikipedia

We used Clearbit Logo's API to search and use company logos for our application. The API is completely free to use with link attribution to Clearbit's website. We chose to display the corresponding logo for each company to enhance our website visualisation.

For company data such as name and description, we decided to use Wikipedia's API. It falls under the Creative Commons licence, allowing us to share and adapt the information retrieved to our application. The API offers results for most, if not all, requests we make, while the documentation is clear and straightforward to use. Therefore, Wikipedia's API is most viable for our application's purpose.

Career Information API:

1. CareerJet

CareerJet's API provides our application with a reasonable number of jobs with each job containing substantial information pertaining to the job requirements and description. The API is also under an MIT licence, placing minimised restrictions on our use of the information we have gathered with the API. Unlike other APIs such as Indeed, Seek and LinkedIn, CareerJet does not require a generated API key to access the data that it provides. Whilst the documentation was scarce and difficult to find, we found that the benefits outweighed this disadvantage and hence decided to use CareerJet's API as the source of our job search.

Location Information API:

1. Google Maps

We decided to use Google Maps' API to show the general location of each job. The API has detailed documentation and we did not require an API key to access the data the API provides. In addition to this, it also has an Apache 2.0 licence, restricting our use minimally.

Authentication API:

1. Google O Auth

The login feature of our application allows for the option to sign in with Google. Thus, we chose to use the Google O Auth API for users to sign in or register with their Google account. Similar to the Google Maps API, the Google O Auth API has extensive documentation and is well-supported for our application and has minimal restrictions.

Design Summary:

Frontend Language Choice

The language used for the frontend is **JavaScript**, using the runtime environment **Node.js**.

Benefits:

1. Simplicity
2. Lower development time as most members is familiar with JavaScript over alternatives like TypeScript.

Cons:

1. Lack of static type definition (No need to declare what type a variable is)

Front-end framework/library

React was chosen as the library for building reusable UI component. React is a frontend JavaScript library which uses JSX, a declarative programming language which is a superset of HTML, allowing injection of JavaScript into it. React uses a virtual DOM, a copy the website's DOM, which is faster to manipulate than the actual DOM. React by itself is minimal in size, as many features are not built into React by default such as routing.

Benefits

1. Uses NPM/yarn for build, test and serving
2. Performant because of using techniques such as the virtual DOM.
3. its huge community, which offsets its weakness of not having many basic features built in, as a huge ecosystem of 3rd party libraries, packages, tools, and extensions exists, with routing handled by the react-router-dom library, state management with Redux, a huge amount of component libraries such as material UI.
4. most team members are familiar with React due to its popularity.

Cons:

1. Does not come with some basic features. (Mitigated by 3rd party tooling)

Component Library

The main UI component library used is **Material UI**, a library based on Google's Material Design which supports the best practices of user interface design. It comes with many reusable components, such as buttons, navigation bars, modals, and so on, and in addition a grid system for layouts.

Benefits:

1. comes with a larger number of components, and a huge number of themes which can be bootstrapped, which is what this project did, and templates which can be combined to form a starter. Allowing us to save a lot of time without having to write as much boilerplate code.
2. Made for react.

Database

For the database, **MongoDB** is chosen in this project. MongoDB is a non-relational database that is schemeless and provides support for JSON-like storage.

Benefits:

1. MongoDB provides more flexibility since different types of data can be saved in a separate JSON document.
2. MongoDB is more friendly to us. It allows us to simply construct a JSON-format document and store the data, which has a smoother learning curve and is suitable for rapid development.

Cons:

1. MongoDB has high memory usage due to the lack of functionality of joins.
2. The maximum size of BSON document (binary representation of JSON documents), where the data records are stored, is 16 megabytes. However, the limitation of MongoDB is acceptable since the scale of this project is small, and it will not involve a large amount of data.

Object Document Mapper

Since non-relational databased has been selected, we would like to use **ODM (Object Document Mapper)** framework. It is a similar technology to ORM (Object Relational Mapper), but it is developed for document databases.

Benefits:

1. Implementing ODM is more efficient. ODM maps a document database system to objects, making data access more abstract and portable, eliminating repetitive code, and allows the implementation of object-oriented design patterns.
2. There are lots of useful libraries that support ODM, such as Mongokit, Pymongo, Mangoose for Python, which largely increases development effectiveness.

Cons:

1. Using ODM may reduce the performance with a huge volume of data, but this drawback is acceptable since this project will not involve a large amount of data.

API Management

To communicate with the backend, API management is important. Our final decision is to implement **Rest API.**

Benefits:

1. All the team members understand how it works and how to implement the framework, boosting our efficiency. Hence, Rest API is considered more adequate for this project by considering the limited time.

Web Server

Lastly, for the backend server, the **Flask** framework that written in Python will be implemented.

Benefits:

1. Python is considered appropriate to use in this project, since all the team members have learnt Python.
2. Flask is light and quick.

3. Flask is that it has a useful extension called Flask-RESTX. Flask-RESTX provides a coherent collection of decorators and tools to document the API used in the project, so that Swagger API documentation can be automatically generated from code, saving time and increasing effectiveness.

Cons:

1. Although Flask does not have many built-in tools, the same functionalities can be achieved by importing modules or libraries.

Part III: Team Organisation:

Team Contribution:

All members agree that everyone contributed equally to this project.

Responsibilities/Organisation:

Kelly:

- Frontend
  - Implemented Statistics page
- Backend
  - Implemented stats organisation in stats.py
- Deliverable 1
  - Contributed ideas for problem statements
  - Developed user stories derived from problem statements
  - Created the Low and High-Fidelity prototype for the "Home" page
  - Styled the High-Fidelity login modals
- Deliverable 2 Report
  - Helped with the System/Architectural diagram
  - Researched and discussed the APIs used in the application
  - Created a few sequence diagrams
- Deliverable 3 Presentation
  - Introduced the main APIs used for the application
- Final Deliverables
  - Updated some of the user stories, a few sequence diagrams and the ER diagram
  - Updated the list of APIs the application is using

Kaivalya:

- Backend
  - Implemented fetching user profiles and adding/modifying job boards
  - Implemented the necessary database function for the above.
- Deliverable 1
  - Contributed to the development of problem statements and user stories
  - Created the Low Fidelity prototype for the "Job search/Job results" page
  - Created High Fidelity FAQ page (not included in final product)
- Deliverable 2 Report
  - Researched for possible design for architectural diagrams, contributed to initial designs.
  - Created the ER diagram
  - Wrote report summary.
- Deliverable 3 Presentation
  - Introduced the team and presented the problem space/business value of the application.
- Deliverable 4/5
  - Introduced the problem space of our application during the presentation.
  - Wrote parts of the deliverable 4 script/slides
  - Updated user stories, rewrote the summary for this report,
  - Wrote the introduction for the business part and listed the main features of our application

Katrina:

- Frontend
    - Implemented the user interface of the Company Profile page.
    - Implemented the user interface of the Job Details page.
- Backend
    - Implemented get, post, delete methods for "/api/user/company" route.
- Deliverable 1
    - Constructed some user stories.
    - Designed low fidelity protype for Job Details page.
- Deliverable 2 Report
    - Justified the technology decisions for application's backend.
    - Updated company profile part for architectural diagram.
    - Constructed UML diagrams and sequence diagrams for some of the pages/components.
- Deliverable 3 Presentation
    - Introduced the technology decisions for application's backend.
- Final Deliverables
    - Constructed UML diagrams and sequence diagrams for some of the pages/components.
    - Updated user stories.
    - Summarized the key technologies used in application's backend and their benefits.

Arthur:

- Frontend
    - Implemented parts of job post cards
    - Worked on parts of job search page, job details and company details
- Backend
    - Data fetching for job details page
    - Implemented company details route and all data fetching involved through external APIs
- Deliverable 1
    - Designed low-fidelity prototype for statistics and settings page
    - Worked on high fidelity prototype, implementing how parts of job search page looked and parts of settings
- Deliverable 2
    - Compared different options for frontend, and justified what was used in the end
    - Made UML diagrams for pages job post search and settings
    - Made sequence diagrams for tailored job recommendations
- Deliverable 3
    - Summarised justification for frontend technology used
- Deliverable 4/5
    - Wrote script for job search and details part of presentation.
    - Updated previous contributions from deliverable 1 and 2
    - Helped cleaning up frontend

Tim:

- Frontend
    - Adapted a React UI template codebase to the application's requirements
    - Implemented job dashboard page and automated job tracking system

    - Implemented job search page querying
- Backend
    - Set up the Flask and MongoDB backend infrastructure including route handler Blueprints and data models
    - Implemented endpoints for job searching, authentication, job tracking and user board management
    - Implemented statistics tracking
- Deliverable 1
    - Designed the low-fidelity prototype for the dashboard page
    - Reworked an existing MIT React template for the team to replicate low-fidelity prototypes
    - Wrote user stories related to job board automation
- Deliverable 2 Report
    - Designed deployment diagram and wrote the deployment technologies section
    - Created UML diagrams for job dashboard
    - Created sequence diagrams related to job board automation
- Deliverable 3 Pitch Presentation
    - Delivered and wrote the script for the live demonstration section
- Deliverable 4 Final Presentation
    - Delivered the live demonstration
- Deliverable 5 Final Report
    - Cleaned up codebase and refactored tightly coupled frontend components
    - Refined the project reflection sections

Project Reflection:

What went well:

1. The team was able to implement all the features that were considered a part of the minimum viable product as well as several extensions that were initially dropped due to them being low priority features.
2. The team almost always met deadlines for tasks that were assigned to each member between each meeting. For instances where a team member was unable to complete a task, the team was able to form an adaptable plan and move past unexpected issues.
3. The team had regular full team meetings, so no member was ever left behind during the term and there were very few issues with miscommunication or lack of participation.
4. There were almost no major design disagreements, the team always communicated their ideas clearly and everyone was encouraged to give their input which helped drive the development of unique features.

What didn't go well:

1. There were no set responsibilities for members of the team. Everyone had been assigned tasks spread over many aspects of the system. The team would have benefited from some clear role distinctions, where for example, someone could have been assigned a specific part of the system.
2. The team was unable to deliver on a few ambitious features that were initially planned due to time constraints. For example, the community system and job recommendation system are quite underdeveloped in the final product.

3. The team didn't make use of team management tools such as Confluence or Jira. Communication was done primarily through instant messaging which was very convenient, however this was not a scalable way for the team to assign tasks and share knowledge during several weeks of challenging planning and development work.

Issues/Problems Encountered:

1. Lack of proficiency in TypeScript, Redux and other technologies and software design practices:
   Initially, the team believed that technologies such as TypeScript, Express, Redux and GraphQL would have been ideal for the project, however each member would have needed to invest an infeasible amount of time to learning these technologies. These early decisions led to a less maintainable and scalable final product.
2. Lack of APIs that supply the required information:
   Major career platforms such as Seek, Indeed and Glassdoor did not readily provide the information this application required. It was necessary to instead use CareerJet's less expansive database of job postings and rely on web scraping techniques to extract the further information that needed to be shown on the front-end.
3. Extremely slow planning phase:
   The team required a very long time to decide on the problem space, develop the vision for the product and the user stories and features that were meant to address the problem statements. Overall, this reduced the amount of time that could be spent on development. The time spent at the start of the project was ineffective because it became evident after starting the project that many requirements needed to be changed to fit in the scope of 10 weeks.
4. Lack of development time:
   With only a few weeks of development time, the team had to drop many ambitious features that were planned at the start of the project. Due to the tight time constraints, the team did very little testing and focused on implementing the features for a minimum viable product rather than focusing on developing a robust and performant system.
5. Online meetings were often laborious and unnecessarily long:
   The team often scheduled a few meetings each week, often lasting between 1-2 hours. The meetings focused on merging work completed since the previous meeting and planning tasks to meet before the next meeting, however they were always very unstructured and sometimes the team was blocked on very small details. For example, a significant amount of time was spent addressing user stories as a group when this could have been divided for each member to handle independently

Improvements:

The following is a list of goals for future software group projects, based on the challenges faced over the last ten weeks for this project:

- Have shorter and more productive meetings that focus on higher-level details and planning and making important decisions rather than coding and writing documents together.
    - Have regular in-person meetings so that it's easier and more natural for the team to interact, exchange ideas and plan.
- Employ more Agile methodologies:
    - Create adaptable plans and work in 1–2-week sprint cycles so that the team can continuously push out new features to production.
    - Regular asynchronous stand-ups to update the team on progress and roadblocks.

    - Clearer division of the team into roles such as ‘developer’, ‘product owner’, ‘project manager’.
    - Follow test driven development principles so that the team can better understand the requirements of the system and deliver robust code.
- Spend more time getting the whole team familiar with the tech stack. Overall, team members were not familiar with modern React features and focused on applying their limited knowledge to solve problems, leading to many ineffective implementations that did not leverage the functionalities offered by React’s APIs. For example, due to a lack of understanding on React hooks, some UI components were implemented using class based React components with deprecated lifecycle methods.
- Choose tech stacks based on what fits the project the best, not on what each member is most comfortable with. Although choosing technologies that the team members were familiar with helped speed up development time, the application’s long-term maintainability and scalability suffered. For example, due to the team's lack of skill in TypeScript, the front-end codebase is extremely fragile, and due to a lack of knowledge in Redux, the application’s state management became increasingly difficult to manage when new features were introduced.
- Use a testing framework for both the frontend and backend codebases such as Jest and Pytest. The system is not robust enough to be deployed safely to a real userbase. Having unit tests, integration tests and end-to-end tests as part of the CI/CD pipeline would have allowed the team to confidently push out new features to production and ensure that the product is reliable at every stage.
- Use Docker for portable deployment from a local development environment to a production environment. The team decided against using Docker due to the significant cost in time in learning this technology.
- Use dedicated team management software such as Confluence or Jira for building up a knowledge base. This would have been helpful for creating a roadmap over the 10 weeks, sharing knowledge gained from solving encountered challenges and keeping every team member informed about the current state of the project.
- Write API documentation before diving into implementation so that team members can independently work on the frontend and backend without having conflicting interpretations of what the inputs and outputs are for a particular endpoint.