**Software Engineering Department**

**Braude College of Engineering**

**MITHABTEY - CareerSeeker**

**Project code: 24-1-D-38**

**Capstone Project Phase B – 61998**

**September 2024**

**Daniel Maman   
 Ben Avner Merigen**

**Git repository link:**[**https://github.com/MamanDaniel/MITHABTEY-CareerSeekers**](https://github.com/MamanDaniel/MITHABTEY-CareerSeekers)

**Supervisor:   
Dr. Naomi Unkelos-Shpigel**

**Table of contents**

[**1. Introduction 5**](#_98mli1f4wx65)

[**2. Project Review 6**](#_p8n0ebgdvqsj)

[**2.1. Working with client 6**](#_y2b128ywrc4o)

[**2.2. Challenges during the development 7**](#_tyjcwt)

[**3. RAMAK questionnaire 8**](#_kybidj2vrzbs)

[**4. Research / Engineering Process 11**](#_ul0dal1hhs4d)

[**4.1. Genetic Algorithm 11**](#_aa9i7hewg8ec)

[**4.2. Genetic algorithm in our site 11**](#_1brqghazsc2k)

[**4.3. Genetic algorithm flow 11**](#_4oomb1kk33jp)

[**4.3.1. Pseudo code 13**](#_2471msm0drsa)

[**4.3.2. Fitness function 14**](#_b718b5ujyqm2)

[**4.3.3. Selection 14**](#_8ap1uwbhuxft)

[**4.3.4. Crossover 16**](#_6c4pdzp04zgl)

[**4.3.5. Mutation 17**](#_36vs72t3g4rt)

[**4.4. Web technologies 18**](#_r2f9pqdp1n4x)

[4.4.1. React 18](#_bjb1fr5k1gk5)

[**4.4.2. TypeScript (TSX) 18**](#_d5rq8b7o2jk5)

[**4.4.3. Node.js 19**](#_g37j0sm2t14w)

[**4.4.4. MongoDB 19**](#_gsdr349jg633)

[**4.4.5. JSON web token - JWT 19**](#_x8y25gl30iod)

[**4.4.6. Deployment of the project 20**](#_mdgeoaw3vlw8)

[**4.4.7. Tailwind 20**](#_zh78d67i61n9)

[**5. Product 21**](#_i1p57ubbf5vu)

[**5.1. Main Interviews during the development 21**](#_3yanxnayqf49)

[**5.1.1. Interview with Irit 21**](#_8z2wzn45bo8h)

[**5.1.2. interview with Ranan Hass 23**](#_31btcon8yi3v)

[**5.2. Requirements 23**](#_7fo9y6kc7x5e)

[**5.2.1. Functional requirements 24**](#_fjh45bhvxnhh)

[**5.2.2. Non-functional requirements 24**](#_3sojogy8onvg)

[**5.3. Preliminary software engineering documents 25**](#_96l8agafqhha)

[**5.3.1. The structure of the system architecture 25**](#_m0zfl661cfr7)

[**5.3.2. Use Case Diagram 25**](#_4iuabq4a27fo)

[**5.3.3. Activity Diagrams 26**](#_ey103kpvcjsw)

[**5.3.3.1. Searching job 26**](#_n69a96m6yauj)

[**5.3.3.2. Adding a new job to the database as an admin 27**](#_a6179ny5jumc)

[**5.4. Testing plan 28**](#_zbzxdctgmu99)

[**6. Expected achievements 30**](#_5tqforbwruy)

[**6.1. Project metrics 30**](#_7i0xubrnrnq0)

[**6.2. Challenges throughout the project 31**](#_r13sfuis1mqr)

[**6.3. Results and conclusions 32**](#_vzjsqqlw6qln)

[**6.4. Lessons learned 32**](#_e1mo1ewcjkk2)

[**6.5. Compliance with setting goals 33**](#_wbm6kcelvzev)

[**7. User Guide 34**](#_1v1yuxt)

[**7.1. Home Page 34**](#_4f1mdlm)

[**7.2. Login Page 35**](#_2u6wntf)

[**7.3. In this page the users can login to the website or register as a new user. 35**](#_1fdktwtchhuw)

[**7.4. Register Page 35**](#_647we0gk8mbf)

[**7.5. Searching Job By Traits Page 36**](#_h187hlpokhoe)

[**7.6. RAMAK questionnaire Page 36**](#_z8wssaj5rqah)

[**7.7. The most 3 suit jobs Page 37**](#_3c16jp5jzxqf)

[**7.8. User`s Profile Page 38**](#_k6nu6oq6c0nx)

[**7.9. Information of jobs that are characterized by RAMAK questionnaire Page 39**](#_dayd8am6j77w)

[**7.10. Admin Panel Page 39**](#_1ukxd2err711)

[**7.10.1. Delete Job Page 40**](#_bs095rh912sx)

[**7.10.2. Users Permissions Page 40**](#_97eic25n7jgr)

[**7.10.3. Update Job Page 41**](#_oiqs402yvjwr)

[**7.10.4. Add Job Page 42**](#_c5mkjdqaw4so)

[**8. Maintenance Guide 43**](#_bcxykdvhl2xe)

[**8.1. General Description 43**](#_vph1t0arg47w)

[**8.2. Database 43**](#_363atr6pqd1)

[**8.3. Usage Scenarios 44**](#_hw2hea3kl5tx)

[**8.3.1. Job Seeker 44**](#_ll0fdzik0gms)

[**8.3.2. Admin 44**](#_qp9vf9wq6j0c)

[**8.3.3. Software Environment 44**](#_r5nd0sb1shzd)

[**8.3.4. Running Instructions 44**](#_ro09xlfepjm)

[**8.3.5. Routing 45**](#_qyhhcocv3fgn)

[**8.3.6. Security 46**](#_qc81w3mdptmc)

[**9. References 47**](#_ufr2rkxyzpjq)

[**10. Appendix 48**](#_dkavreiluw6k)

[**10.1. Questions for interviews in the Facebook group 48**](#_66xfrzpq7pfw)

[**10.2. Meeting schedule with client - Irit Hommasi**](#_ahzjrk8pwe4) **48**

**Abstract**

In today's world, many people are looking for the ideal career path that aligns with their interests and skills. This decision is crucial as it plays a significant role in shaping their future prospects. However, the process can be challenging due to the vast array of job opportunities available and the limited information accessible about each one. Typically, people gather insights about various professions through discussions with friends, conducting online research, or connecting with acquaintances who are already working in a particular field.

In today's world, many people are looking for the ideal career path that aligns with their interests and skills. This decision is crucial as it plays a significant role in shaping their future prospects. However, the process can be challenging due to the vast array of job opportunities available and the limited information accessible about each one. Typically, people gather insights about various professions through discussions with friends, conducting online research, or connecting with acquaintances who are already working in a particular field.  
Also,many new career paths are evolving on a daily basis.

This project is intended for a facebook group called 'Mithabtey Miktzoa' (Job seekers) which has 40,000 members who are looking for a solution for this problem.

In our project we utilize genetic algorithm to help the group members in 'Mithabtey Miktzoa' to find jobs that match their character traits in the easiest way.

**Key words:** Character traits, jobs, genetic algorithm, Website development, RAMAK questionnaire.

# 

# 1. Introduction

In today's job market, finding the right career path has become increasingly challenging. With the emergence of new job opportunities and a growing complexity in job roles, individuals often face difficulty in identifying suitable professions.  
Not only is it challenging to determine which profession you want, but there are also professions that you may not be aware of but might interest you.  
Some people find work with recruiters who analyze their social media profiles but this is ineffective[[2](#_5tqforbwruy)].  
To address this challenge, a Facebook group named "Mithabtey Miktzoa" (translation: "My Job, My Profession") has been established, managed by Irit Hommsi. The primary objective of this online community is to bridge the information gap by providing its members with valuable insights and details about various unique and lesser-known career opportunities [[12](#pvoj6khn4x)].

The group serves as a platform for individuals to explore a wide range of professions, some of which may not be commonly discussed or well-known. By sharing comprehensive information about these unique job prospects, "Mithabtey Miktzoa" aims to empower its members to make well-informed decisions about their future career paths.

Through this Facebook group, members can access a wealth of resources, including detailed job descriptions, educational requirements, necessary skills, and potential growth opportunities within each profession. Additionally, the group facilitates discussions and Q&A sessions with professionals from diverse fields, allowing members to gain first-hand knowledge and guidance from those with real-world experience. Examples for such questions:

* What does a standard working day at work look like?
* How do you start working?
* What technical skills are necessary?
* What experience do you need to get hired?

and more (see appendix [10.1](#8uzkqruusp2u))  
After choosing this idea for our final project, we were tasked with developing a system for this process of career exploration. The system will operate through a website accessible to all users. Users can complete the RAMAK questionnaire to discover their personality traits. The system will match between users' traits with suitable jobs using a genetic algorithm that collects data from the Facebook page. This group's information is extracted, organized, and then integrated into our database.

# 2. Project Review

To build this web application, we employed a range of modern technologies chosen for their efficiency, scalability, and user-friendliness:

* Backend: We utilized technologies such as Node.js to create a robust server-side architecture. This allowed us to implement the genetic algorithm efficiently and manage the database of job opportunities.
* Frontend: For the client-side, we employed a modern TypeScript framework such as React. These technologies enabled us to create a responsive and interactive user interface, ensuring a smooth experience for users as they navigate through the RAMAK questionnaire and view their job matches.
* Database: We implemented a suitable database system MongoDB to store and manage the job data collected from the 'Mithabtey Miktzoa' Facebook group.
* API: We developed a RESTful API to facilitate communication between the frontend and backend, ensuring seamless data flow and real-time updates.

## 2.1. Working with client

We maintained regular communication and collaboration with the client throughout the development process, ensuring their requirements were thoroughly addressed. In part A of the project, we implemented the specifications provided by the client, while also adapting to any evolving needs as the project progressed. This close partnership was facilitated through frequent WhatsApp conversations and Zoom meetings, allowing for real-time feedback and discussions that ensured the project stayed aligned with the client's vision.

During development, we requested and received various documents from the client that were crucial for managing different aspects of the project. These documents included detailed project requirements, user workflows, and specific feature requests, which provided invaluable insights into the client’s needs. Having these documents as a reference greatly streamlined our development process, helping us tailor the application more accurately to meet their expectations and objectives. The client’s ongoing input and documentation allowed us to iterate effectively and deliver a solution that truly reflected their goals.

## 

## 2.2. Challenges during the development

1. Our client was not available for a long time, which created delays in the development of the project.
2. The system we need to build should contain many features that the client requires.
3. Some features are easy to implement and some are more difficult and logically complicated.
4. The system should be reactive and easy to use.
5. The system should be used by a wide variety of users and will have different types of users.
6. The different types of users have different permissions, so this is a big challenge in itself in implementing the system.
7. We have two entities in our system: user and admin.
8. Anyone can register for our application, but only the admin can give permissions to the user to be an admin.
9. We had to maintain the protection of certain features, only at the level of permission of an admin, the user will be able to do the actions. And similarly, there are features that only the admin can do.
10. Another challenge is working on 2 different main platforms, one on the server side and the other on the client side. In order to overcome this challenge and to avoid confusion and waste of time, we decided to split the work so that one of us will work on the server side and the other on the client side, thus we can work simultaneously without being dependent on each other until the integration stage of the features.
11. Another challenge we faced was working with a database. Both on the server side and on the client side we had to work against a database such as creating new users, creating project proposals and giving approvals on the server side. And on the client side, retrieving data in order to use it, saving data that is received from the users.

# 3. RAMAK questionnaire

Anne Roe, a clinical psychologist, developed an influential theory categorizing occupations into eight groups based on their required interaction with people versus non-people environments.  
Her theory proposed that early childhood experiences and psychological needs shape an individual's orientation toward preferring interpersonal or impersonal settings, which then guides their career choices. Roe's work provided a framework linking psychological development in childhood to eventual vocational preferences and career decisions.  
The RAMAK questionnaire, developed in 2011 by Elhanan Meir, is an assessment tool grounded in Anne Roe's Occupational Classification theory, which categorizes individual interests across eight distinct domains:   
Business, Organization, General Culture, Service, Arts and Entertainment, Outdoor, Science, and Technology.

The Ramak Interest Inventory is a tool designed to assess an individual's interests based on Roe's classification, aiding in career exploration and decision-making. Ramak also discusses the significance of understanding one's interests in career planning and how tools like the Ramak Inventory can assist in this process[[8](#kix.qaqox2so4vtr)].

The RAMAK Questionnaire typically includes a set of questions designed to measure:

1. **Business:** This category includes interests related to business operations, management, finance, and entrepreneurship. People interested in business may enjoy roles such as business administration, finance management, marketing, or human resources.
2. **Organization:** Interests in this category involve activities related to organizing, planning, and coordinating. Careers in project management, event planning, logistics, or administration may appeal to those with a strong organization interest.
3. **General Culture:** Interests in general culture involve a curiosity about the arts, history, literature, and other aspects of human civilization. Careers in education, museum curation, journalism, or cultural preservation may appeal to individuals with a strong general culture interest.
4. **Service:** This category includes interests related to helping, assisting, and caring for others. People interested in service may enjoy roles such as teaching, counseling, healthcare, or social work.
5. **Arts and Entertainment:** This category includes interests in artistic expression, creativity, and entertainment. People interested in arts and entertainment may pursue careers in music, theater, film, visual arts, or writing.
6. **Outdoor:** Interests in outdoor activities, nature, and the environment fall into this category. Careers in forestry, environmental science, wildlife conservation, or outdoor recreation might be attractive to individuals with a high interest.
7. **Science:** This category encompasses interests in scientific inquiry, research, and experimentation. People interested in science may pursue careers in fields such as biology, chemistry, physics, or environmental science.
8. **Technology:** Interests in technology involve a fascination with computers, electronics, and technological innovations. Careers in software development, IT management, cybersecurity, or digital marketing may be appealing to individuals with a strong technology interest.

**The values of the questionnaire**

The questionnaire contains 72 questions to which the answers are yes no or unknown. For each question a score is obtained in the following way

Score 2 points for a Yes, 1 point for a question mark, and 0 for a No [[4](#kix.qaqox2so4vtr)].

The question key is represented as follows:

1. **Level 1:** typically indicates a high level of interest in that category. Items listed under Level 1 are those that strongly align with the corresponding interest category.
2. **Level 2:** suggests a moderate level of interest. These items are somewhat related to the interest category but may not indicate as strong a preference as Level 1 items.
3. **Level 3:** indicates a lower level of interest. Items at this level are less strongly associated with the interest category and may suggest a less interest.

**Business:** Level 1-Items 10,46,55; Level2-items 21, 29, 37;

Level 3-ltems 3, 60, 69

**General Culture:** Level 1-Items 22, 31, 52; Level2-Items 5,

11,47; Level3-Items 36, 63, 72

**Arts & Entertainment:** Level 1-Items 4, 28, 53; Level2-Items

14,35,61; Level3-items 24, 45, 68

**Science:** Level 1-Items 8, 16,58; Level2-Items 23, 26, 44; Level

3-items 33, 54, 66

**Organization:** Level 1-Items 13, 19,67; Level2-items 8, 38,

41; Level 3-items 12, 30, 42

**Service:** Level 1-Items 1,34,49; Level2-items 20, 59, 65; Level

3-items 12, 30, 42

**Outdoor:** Level 1-Items 39, 64, 70; Level 2-ltems 2, 32, 51;

Level3-Items 3, 9, 18

**Technology:** Level 1-Items 25, 40, 43; Level 2-ltems 17, 50,

71; Level 3-Items 6, 15, 62

We decided to use the RAMAK questionnaire for the job matching system.  
The reasons behind our decision:

* Ra'anan Hess, a psychology professor specializing in employment psychology, recommended the RAMAK questionnaire during an interview.
* The RAMAK questionnaire allows division of traits to three levels for each representative trait.
* The article on the questionnaire contains the questions that examine the person and classify him into the relevant categories, and as a result the questionnaire is valid.
* The RAMAK questionnaire is the most recent among the options considered.

# 4. Research / Engineering Process

## 4.1. Genetic Algorithm

Genetic algorithms are a type of optimization algorithm inspired by the process of natural selection. They are used to find optimal solutions to complex problems by mimicking the process of evolution. In a genetic algorithm, a population of candidate solutions to the problem is evolved over generations through processes such as selection, crossover, and mutation. Each candidate solution, often called an individual, is represented as a set of parameters or genes. These individuals are evaluated using a fitness function, which determines how well they solve the problem. Through successive generations, the population evolves, and better solutions are discovered. Genetic algorithms are particularly useful for optimization problems where traditional methods are impractical or inefficient [[1](#kix.otjceswkjrt)].

## 4.2. Genetic algorithm in our site

Our website aims to effectively match users with professions that suit their character traits. It employs a genetic algorithm that analyzes the user's character data from the RAMAK questionnaire. The algorithm evolves a population of candidate profession matches, represented as combinations of required character traits. Each candidate's fitness is evaluated by comparing the user's traits to the profession's requirements. Fitter candidates (better matches) are favored to reproduce to the next generation. Variations are introduced through crossover and mutation to explore new trait combinations. Over successive generations, the population converges towards optimal profession matches for the user. Upon convergence, the algorithm outputs a ranked list of professions ordered by their suitability for the user based on their unique character profile. This list empowers users to make informed career decisions aligned with their intrinsic traits.

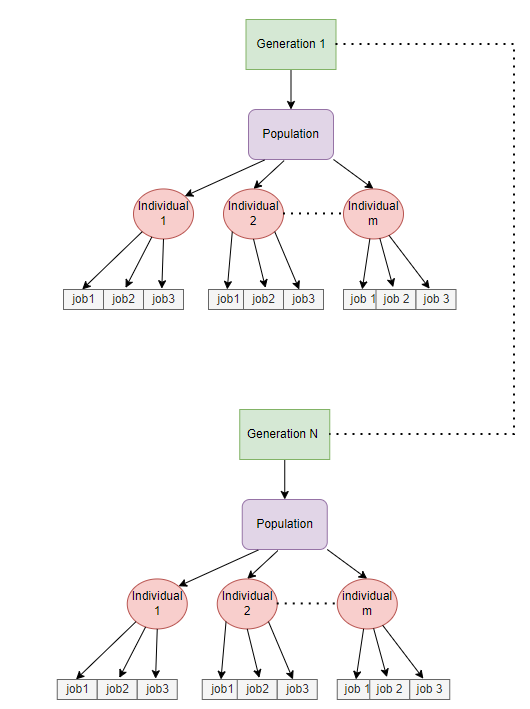
## 4.3. Genetic algorithm flow

The genetic algorithm begins by initializing a population of individuals, where each individual represents a potential solution to the problem. In this case, each individual is a list of a predefined amount of random professions. The algorithm then evaluates the fitness of each individual by calculating the average match percentage between the person's traits and the traits required for each profession.

Next, the algorithm enters a loop where it selects two parents from the population based on their fitness scores using a method called roulette wheel selection. It then performs a one-point crossover between the two parents to produce two offspring. Crossover involves combining genetic information from the parents to create new individuals. After crossover, the algorithm mutates the offspring by randomly changing one of their professions.

The algorithm continues this process for a specified number of generations, maintaining a population size that is defined in advance. In each generation, the top two individuals from the previous generation are retained (elitism), and the rest of the population is replaced with new individuals generated through crossover and mutation.

After the specified number of generations, the algorithm evaluates the final population and extracts the top three unique professions from the final sorted population. These professions are considered the best match for the user based on their character traits.

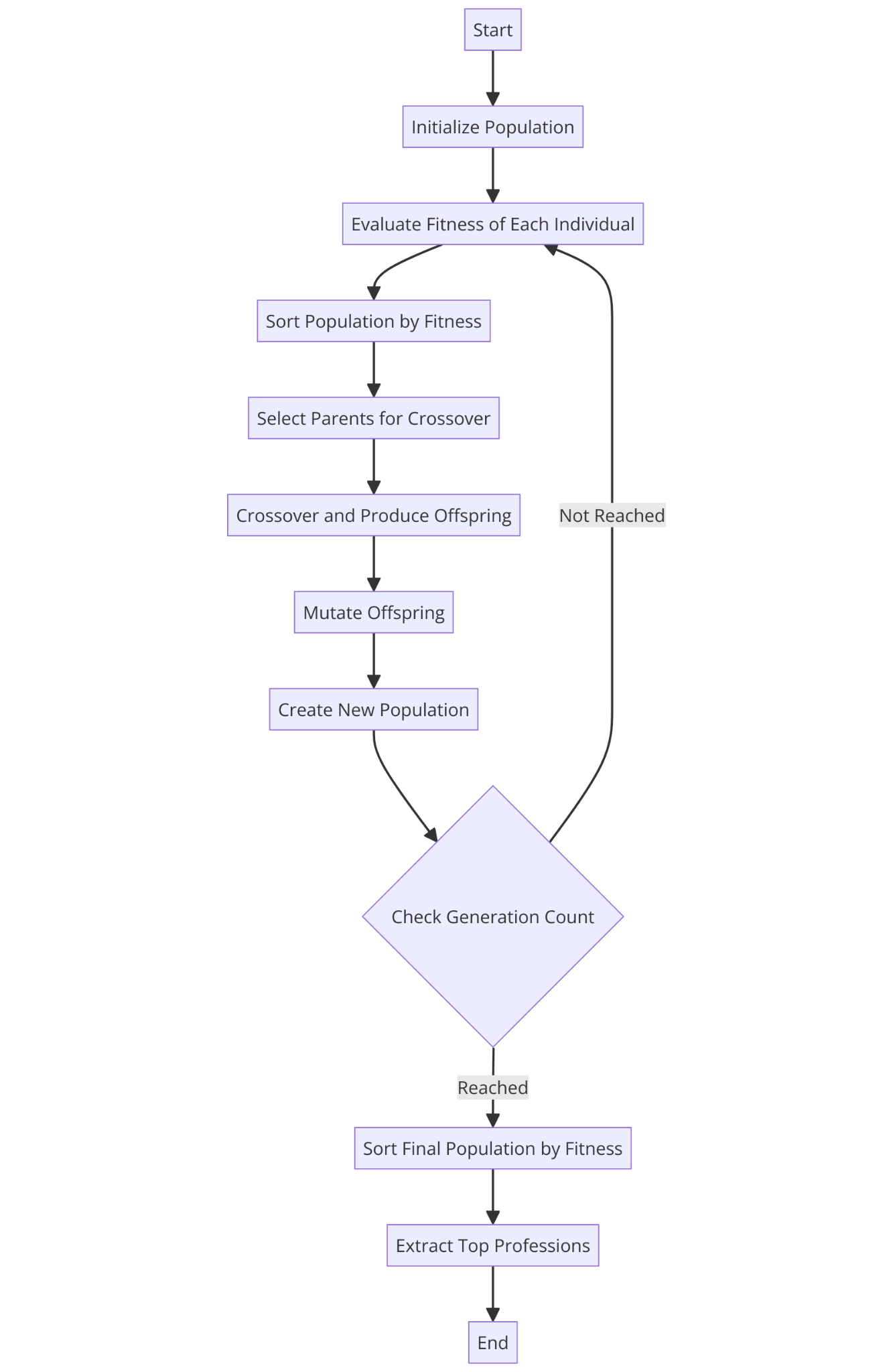
  
*Figure1: Our genetic algorithm hierarchy*

### 

### 4.3.1. Pseudo code

analyze\_user\_traits\_by\_RAMAK\_questionnaire()  
 For each question in questionnaire  
 responses += get\_user\_choose  
 analyze\_by\_ramak(responses)  
 return user\_traits\_by\_ramak

genetic\_algorithm(user\_traits\_by\_ramak,list\_of\_jobs,num\_generations,population\_size)  
 Population = generate\_first\_generation()  
 While (num\_of\_generations)  
 Population\_fitness = fitness(Population)  
 parent\_1, parent\_2 = selection(Population ,Population\_fitness)  
 New\_generation = crossover(parent\_1, parent\_2)  
 New\_population = mutation(New\_generation)   
 Population = New\_population   
 Return population

  
*Figure2: Diagram illustrating the flow of a genetic algorithm*

### 4.3.2. Fitness function

Fitness is a key concept in genetic algorithms, representing how well an individual in the population solves the problem at hand. In our code, the `evaluate\_individual` function calculates the fitness of each individual by measuring the match percentage between the person's traits and the traits required for each of the professions in the individual's list. This match percentage is then used as a measure of the individual's fitness.

`evaluate\_individual` function takes an individual (a list of profession indices), the person's traits, and a list of professions with their traits. It then iterates over the professions in the individual, calculates the match percentage for each profession by counting the number of matching traits, and computes the average match percentage for all professions in the individual. This average match percentage represents the fitness of the individual.

For example, if an individual has professions that require traits similar to the person's traits, it will have a higher fitness score. Conversely, if an individual's professions do not match well with the person's traits, it will have a lower fitness score. This fitness score guides the selection process in the genetic algorithm, favoring individuals with higher fitness scores for reproduction and potentially leading to better solutions over time as the algorithm progresses.

# Calculate the fitness of an individual  
fitness\_of\_individual(individual, user\_traits\_by\_RAMAK, list\_of\_jobs):

match\_percentages = []

for job\_index in individual:

job = list\_of\_jobs[job\_index]

match = calculate\_match\_percentage (user\_traits\_by\_RAMAK, job['traits'])

match\_percentages.append(match)

return sum(match\_percentages)/len(individual)

# Calculate the match percentage between a person's traits and a profession's traits

calculate\_match\_percentage (user\_traits\_by\_RAMAK, job\_traits):

match = sum([1 for trait in person\_traits\_by\_RAMAK if trait in job\_traits])   
 return match / len(job\_traits) \* 100

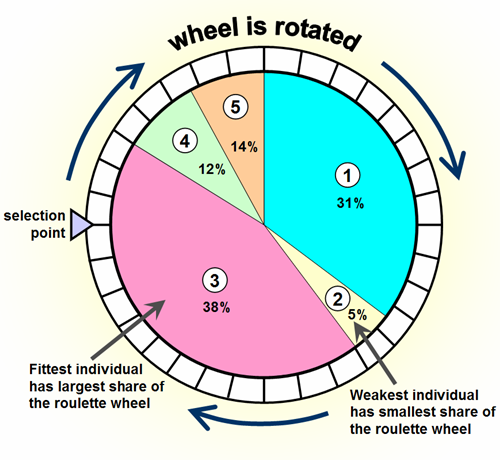
### 4.3.3. Selection

The foundational aspect of the selection process is to stochastically select individuals from one generation to create the basis for the next. The key requirement is that the fittest individuals possess a greater probability of survival compared to weaker ones. This emulates nature, where fitter individuals tend to have an increased likelihood of survival and advancement to form the mating pool for subsequent generations. However, weaker individuals are not entirely excluded. In the natural world, such individuals may harbor genetic coding that could prove advantageous for future generations, contributing to the overall diversity and adaptability of the population.  
In our code, the `select\_parents` function implements selection using a method called roulette wheel selection. This method selects individuals with a probability proportional to their fitness scores, giving fitter individuals a higher chance of being selected as parents.

The `select\_parents` function takes the population and their corresponding fitness scores as input. It calculates the total fitness of the population and then computes the selection probabilities for each individual based on their fitness scores. These probabilities are used to select two parents randomly from the population, with fitter individuals having a higher probability of being chosen.

For example, if an individual has a high fitness score, it will have a higher probability of being selected as a parent. Conversely, individuals with lower fitness scores will have a lower probability of being selected. This selection process helps maintain diversity in the population while favoring individuals with better fitness, allowing the genetic algorithm to explore promising areas of the search space and potentially find optimal solutions to the problem.

# Select two parents from the population based on their fitness scores select\_parents(population, fitness\_scores):  
 total\_fitness = sum(fitness\_scores)  
 selection\_probs = [score / total\_fitness for score in fitness\_scores]  
 return random.choices(population, weights=selection\_probs, k=2)

  
*Figure3: Diagram illustrating the probability of select parents [*[*29*](#i1i0af34d8h5)*]*

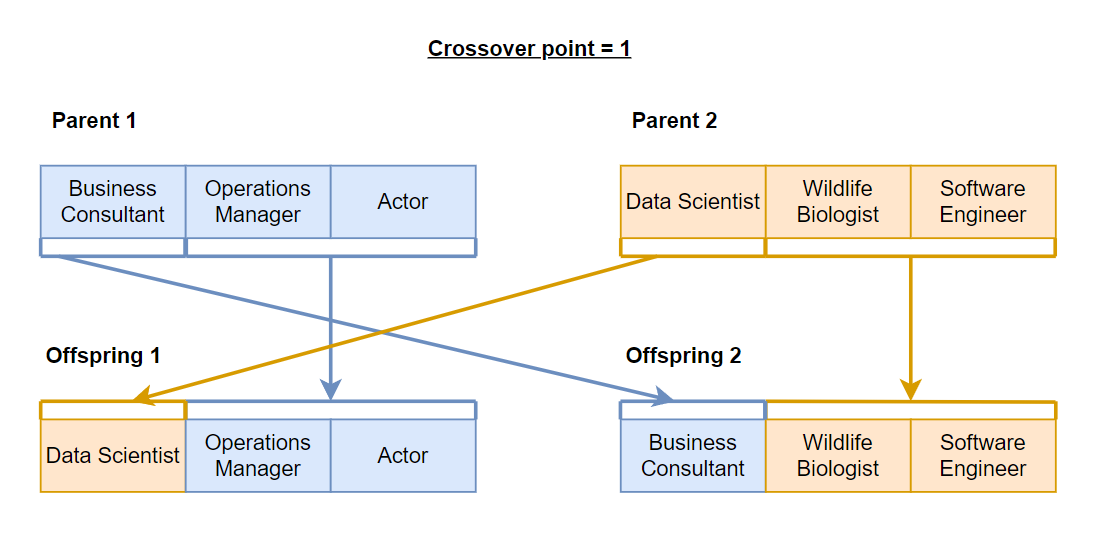
### 4.3.4. Crossover

Crossover, also known as recombination, is a genetic operator used in genetic algorithms to combine genetic information from two parent individuals to create new offspring. In our code, the `crossover` function implements one-point crossover, a common method where a random crossover point is selected along the length of the parents' chromosomes (in this case, the list of professions). The function then creates two offspring by combining the genetic information from the parents at the selected crossover point.

In our code, the `crossover` function takes two parents (`parent1` and `parent2`) and randomly selects a crossover point (`crossover\_point`) between the first and second-last elements of the parents' lists. It then creates two offspring (`offspring1` and `offspring2`) by combining the genetic information from the parents: `offspring1` consists of the first part of `parent1` up to the `crossover\_point`, followed by the second part of `parent2` from the `crossover\_point` onwards. Similarly, `offspring2` consists of the first part of `parent2` up to the `crossover\_point`, followed by the second part of `parent1` from the `crossover\_point` onwards.

This process of combining genetic information from two parents through crossover helps maintain genetic diversity in the population and can lead to the discovery of new, potentially better solutions to the problem at hand.

# Perform one-point crossover between two parents to produce two offspring  
crossover(parent\_1, parent\_2):  
 crossover\_point = random.randint(1, len(parent\_1) - 2)  
 offspring\_1 = parent\_1[:crossover\_point] + parent\_2[crossover\_point:]  
 offspring\_2 = parent\_2[:crossover\_point] + parent\_1[crossover\_point:]  
 return offspring\_1, offspring\_2

  
 *Figure4: Example of crossover process*

### 4.3.5. Mutation

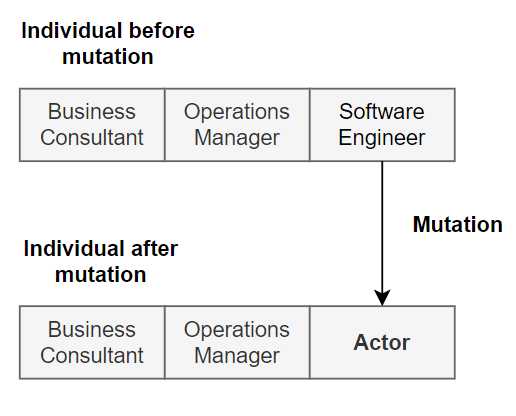
Mutation is a genetic operator used in genetic algorithms to introduce random changes in individuals' genetic information.

The function implements mutation by randomly changing one of the professions in an individual. This random change helps maintain genetic diversity in the population and can prevent the algorithm from converging to a local optimum.

The function takes an individual (represented as a list of professions) and randomly selects a profession to mutate. It then changes the selected profession to a random profession index. This random change introduces variability in the population, allowing the algorithm to explore different solutions in the search space.

The mutation is applied to offspring after crossover, ensuring that each new generation of individuals has some level of genetic diversity [[3](#kix.nlilckgbmuvd)]. This diversity is crucial for the genetic algorithm to explore a wide range of potential solutions and find the optimal or near-optimal solution to the problem.

# Mutate an individual by randomly changing one of their professions  
mutate(individual, num\_of\_jobs):  
 mutation\_index = random.randint(0, len(individual) - 1)  
 individual[mutation\_index] = random.randint(0, num\_of\_jobs - 1)  
 return individual

  
*Figure5: Example of mutation process*

## 4.4. Web technologies

**Introduction**

Developing a MERN website involves building a full-stack web application using MongoDB, Express.js, React, and Node.js. MongoDB serves as the database to store and retrieve data, offering flexibility with its NoSQL structure. Express.js, a backend web framework, works with Node.js to handle server-side logic, APIs, and requests. React is used to create a dynamic, interactive front end, enabling a seamless user interface. Node.js, which powers the server, manages non-blocking, event-driven operations to handle multiple requests efficiently. Together, these technologies offer a robust platform for creating modern, scalable web applications with a JavaScript-driven stack.

### 4.4.1. React

React, a popular JavaScript library developed by Facebook for building user interfaces, especially single-page applications (SPAs), emphasizes the use of reusable components, making it efficient for developing complex interfaces with dynamic content. It optimizes rendering through a virtual DOM, updating only what's necessary to improve performance. With its component-based architecture, React promotes modularity and maintainability, while JSX allows developers to write HTML-like syntax within JavaScript, enhancing readability. React's hooks enable functional components to manage state and side effects, and its experimental Concurrent Mode aims to further improve rendering performance for complex applications.[[6](#1rx6qh5tggnx)]

### 4.4.2. TypeScript (TSX)

TypeScript is a superset of JavaScript developed by Microsoft, adding static typing to the language. It helps catch errors early in development, providing better tooling and enhancing code quality and maintainability.

The static typing allows developers to specify types for variables, functions, and objects, catching type-related errors at compile time. It offers advanced object-oriented programming features, including class and interface support, making it suitable for large-scale application development. Additionally, TypeScript code compiles to plain JavaScript, ensuring compatibility with existing JavaScript codebases and libraries.

In our website we will use TypeScript 4.0 to use features like variadic tuple types, labeled tuple elements, and improved type inference, among others.[[11](#kzhu1h2ocxj0)]

### 

### 4.4.3. Node.js

Node.js is an open-source, cross-platform JavaScript runtime environment that allows developers to execute JavaScript code on the server-side. It's built on Chrome's V8 JavaScript engine and enables the development of scalable network applications [[5](#yuq33bsf61z8)].

Key Features:

1. Asynchronous and Event-Driven: Handles multiple connections concurrently without blocking, making it suitable for building scalable applications.
2. NPM (Node Package Manager): Provides access to a vast library of packages, simplifying the addition of functionalities to applications.
3. Single Programming Language: Allows developers to use JavaScript for both client-side and server-side development, promoting code reusability and efficiency.

### 4.4.4. MongoDB

MongoDB is a NoSQL database that provides high performance, high availability, and easy scalability. It uses a document-oriented data model, making it flexible to accommodate changes in data structure [[9](#ys71k6o5h4m3)].

Key Features:

1. Document-Oriented Storage: Stores data in JSON-like documents, allowing for varied data structures and dynamic schemas.
2. Scalability: Supports horizontal scaling through sharding, enabling the distribution of data across multiple servers.
3. Aggregation Framework: Provides powerful tools for data aggregation and analysis.

### 4.4.5. JSON web token - JWT

**Overview**

JSON Web Token (JWT) is a compact, URL-safe means of representing claims to be transferred between two parties. It is often used for secure authentication and information exchange in web applications. JWTs consist of three parts: a header, a payload, and a signature. The header typically specifies the token type and the algorithm used for signing. The payload contains the claims, which are statements about an entity (usually the user) and additional data. The signature is used to verify that the sender of the JWT is who it says it is and to ensure that the message was not changed along the way.

In our project, JWT integrated for user authentication and session management. When a user logs in, the server can generate a JWT containing the user's information and send it back to the client. The client can then store this token and send it in the Authorization header of subsequent requests to access protected routes. On the server side, middleware can be used to verify the token's signature and extract the user's information from the payload, granting or denying access based on the token's validity. This approach provides a stateless authentication mechanism, as the server does not need to keep track of active sessions, and it scales well for distributed systems, such as those deployed on cloud platforms [[7](#olah2p18nbz0)].

### 4.4.6. Deployment of the project

Deploying a web application involves several key steps to ensure that our website is accessible to users on the internet. Initially, we need to choose a hosting provider or cloud platform, such as AWS or Vercel, which offers the infrastructure and services required to host our application.

Next, we should configure our domain name, which involves registering a unique domain and pointing it to our hosting provider's servers. This step is crucial for making our website easily accessible via a user-friendly URL.

The hosting provider will then build and serve our application, making it available to users worldwide.

**Render**

Web service deployment in Render involves hosting and managing applications seamlessly in the cloud. Render provides a simple platform for developers to deploy web services, whether they are static sites, dynamic applications, or APIs. The process typically starts by connecting a GitHub or GitLab repository, which triggers automatic builds and deployments whenever changes are pushed. Render supports a variety of programming languages and frameworks, and it offers built-in SSL, custom domain management, and scalability options. With features like automatic scaling, continuous deployment, and easy environment variable management, Render makes deploying web services straightforward and efficient, allowing developers to focus on their application's functionality rather than infrastructure management.[[8](#7ojvqg5qlpzu)].

### 4.4.7. Tailwind

Tailwind CSS is a utility-first CSS framework that provides a set of predefined classes, allowing developers to build custom designs directly in their HTML markup. It emphasizes rapid development and encourages a design-first approach, making it easier to create responsive and visually consistent interfaces. By using Tailwind, we can avoid writing repetitive CSS code, leading to cleaner and more maintainable codebases. The framework is highly customizable, enabling users to define their own themes and design systems. Tailwind is high-quality and efficient thanks to its flexibility, ease of use and ability to integrate seamlessly with development tools and modern front-end frameworks [[10](#38ml3o2u7c9b)].

# 5. Product

## 5.1. Main Interviews during the development

### 5.1.1. Interview with Irit

We started our research by talking to Irit, who manages a Facebook page where interviews are conducted with people who come to present their work.  
During these interviews, the interviewees typically face a set of common inquiries. These include questions about the personality traits required for the job, steps to embark on a career in that field, necessary qualifications or prerequisites, and potential earnings. Our initial interview focused on gaining insights into Irit's work revolving around the "Mithabtey Miktzoa" Facebook page and her aspirations for this endeavor. The following is a synopsis of our conversation, highlighting the key points discussed and the primary lessons learned.  
There are 302 professionals that have been interviewed for the website project to match individuals' qualities to suitable professions.  
Currently, the information is on Facebook covering 80 professions, but significant effort is needed to extract and organize it. There are essential tasks that need to be done.

Website requirements:

* Admin panel for updating information
* Algorithm to calculate matches even with incomplete user data
* User-friendly interface to guide trait selection and view matched professions

Additional features:

* User account creation
* Connection to courses and discounts
* Partnership with Find Me Mentor site for mentorship after job matching

In phase B, we tried to implement the ideas by maintaining a dialogue and follow-up with the client on a monthly basis, thus knowing that we are meeting her expectations and requirements . [[interviews dates](#pk71k275qtgx)]

However, the process encountered several unforeseen delays due to various factors such as the health condition of our client, the site manager, Irit, and also due to events that happened in her and our schedule.  
These factors influenced and even delayed the development process.  
Finally, we have successfully concluded the presentation of the website to the client. Following a comprehensive review, the client has granted their approval and expressed satisfaction with the final product.

To match between character traits and professions:

* Use the Big Five Model (FFM) of personality traits for classification
* Extract and organize information from Facebook group comments
* Use Facebook API to extract posts
* Allow users to select and arrange their traits.
* Job matching algorithm to find and rank solutions based on trait and background match
* Organize information in a Google Drive Excel sheet by profession and category.   
  Example of how the information organized in jobs excel:

*Figure6: Example of trait from Irit`s jobs excel*

Figure 6 presents an unstructured Excel file containing information about various professions, which was obtained from Irit. For each profession, the file displays the questions (represented in columns) that were posed to the interviewees, reflecting a set of casual inquiries. The information shown is derived from the general responses provided by the interviewees. However, the data crucial for our algorithm's implementation is the identification of the RAMAK traits that characterize each profession. This information is essential to facilitate the matching process between the user's unique traits and suitable professions, leveraging the genetic algorithm. Extracting these requisite traits from the unstructured data presents a considerable challenge, as there is no discernible pattern or routine that allows for efficient and convenient data mining procedures.

### 5.1.2. interview with Ranan Hass

In our consultation with Ra'anan Hess, an occupational psychologist specializing in job placement, several salient conclusions were drawn regarding the alignment of personality traits with suitable professions. Firstly, Hess elucidated the limitations of the Big Five personality trait model, describing it as "very theoretical" and necessitating extensive questionnaires, which may prove impractical for the purposes of the website. Secondly, while the Holland questionnaire aligns with the concept of matching traits to occupations, it is considered outdated, and the professions it encompasses may not be relevant to contemporary job markets. Lastly, Ra'anan recommended the RAMAK questionnaire, which tends to focus more on work domains rather than solely on character traits, suggesting that it might be the optimal solution for the website's matching algorithm.

The recommendation to utilize RAMAK was based on several factors:

1. RAMAK employs a three-level scale to describe each trait, providing a more nuanced assessment.
2. The questionnaire has been validated and proven through empirical research.
3. RAMAK represents the most up-to-date questionnaire available.

By incorporating Hess's expert guidance and employing the RAMAK questionnaire, the website can potentially enhance its ability to match individuals with suitable career paths based on their unique personality traits and work preferences.

## 5.2. Requirements

In alignment with the customer's specifications and the mandated system architecture, we have conducted a comprehensive analysis to delineate the system's outline, encompassing both functional and non-functional requirements. This rigorous assessment ensures that the proposed solution adheres to the client's expectations and meets the desired objectives. The findings are as follows:

### 5.2.1. Functional requirements

1. The system allows registration.
2. The system allows login.
3. The system allows search jobs by it`s field.
4. The system allows users to receive information about jobs.
5. The system allows for matching between the user`s traits to their most suit job   
   by genetic algorithm.
6. The system allows users to edit their profile.
7. The system allows the admin to add / delete / edit jobs.
8. The system allows the admin to send notifications to users.
9. The system allows users to answer the RAMAK questionnaire.
10. The system will alert users if there are new jobs.
11. The system allows giving feedback on the results obtained from the algorithm.
12. The system allows users to save favorite job listings.
13. The system allows users to connect with mentors or industry professionals for career advice and guidance.
14. The system allows users to participate in forums or discussion boards related to job searching and career development.
15. The system allows adding a new post directly to the Facebook group 'Mithabtey Miktzoa' through the website.
16. The system allows users to apply for jobs directly through the website.

### 5.2.2. Non-functional requirements

1. **Usability**: the website should be easy to navigate and use, with clear instructions for completing the survey and viewing results.
2. **Protability**: The system can be used on different platforms/browsers.
3. **Authentication**: To login you must enter username and password.
4. **Management**: Only admin can add/delete/edit jobs.
5. **Scalability**: The website should be able to handle a large number of users simultaneously without significant performance degradation.
6. **Security**: User data should be securely stored and transmitted, with measures in place to protect against unauthorized access.
7. **Scalability:** The algorithm should be designed to easily scale to growth in the number of users and jobs.
8. **Compatibility**: The website should be compatible with a range of browsers and devices to ensure a seamless user experience.
9. **Maintainability**: The website should be easy to maintain and update, with clear documentation for future development.
10. **Performance:** The website should load quickly and respond to user actions.
11. **Accessibility**: The system design will accommodate people with disabilities.

## 

## 5.3. Preliminary software engineering documents

### 5.3.1. The structure of the system architecture

### 

*Figure7: Architecture scheme [*[*13*](#aism5lbztxnw)*]*

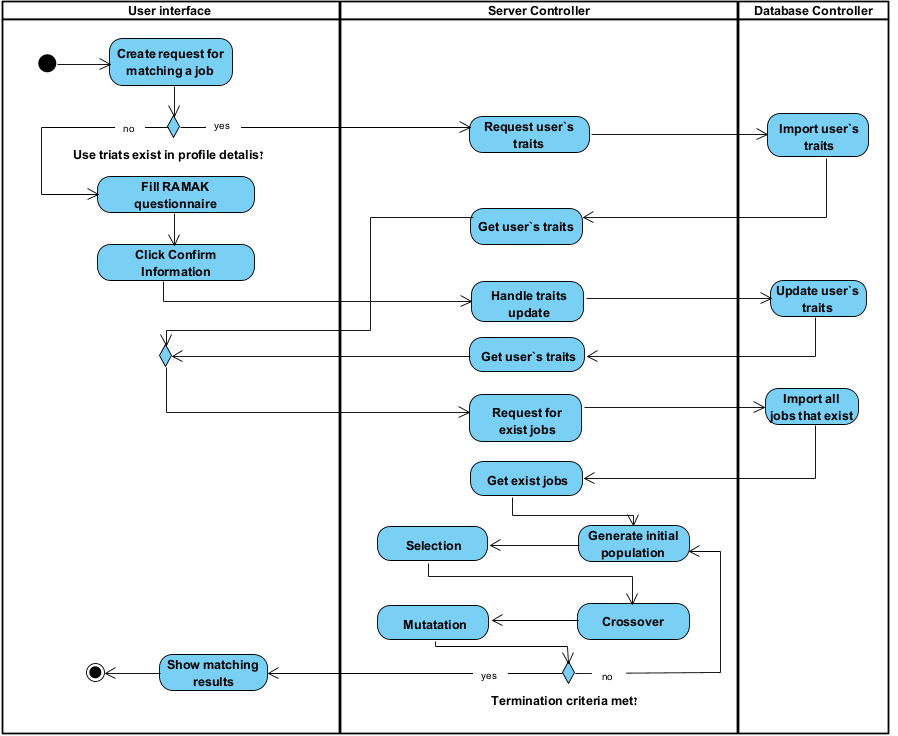
### 5.3.2. Use Case Diagram

### 

*Figure8: Use Case Diagram*

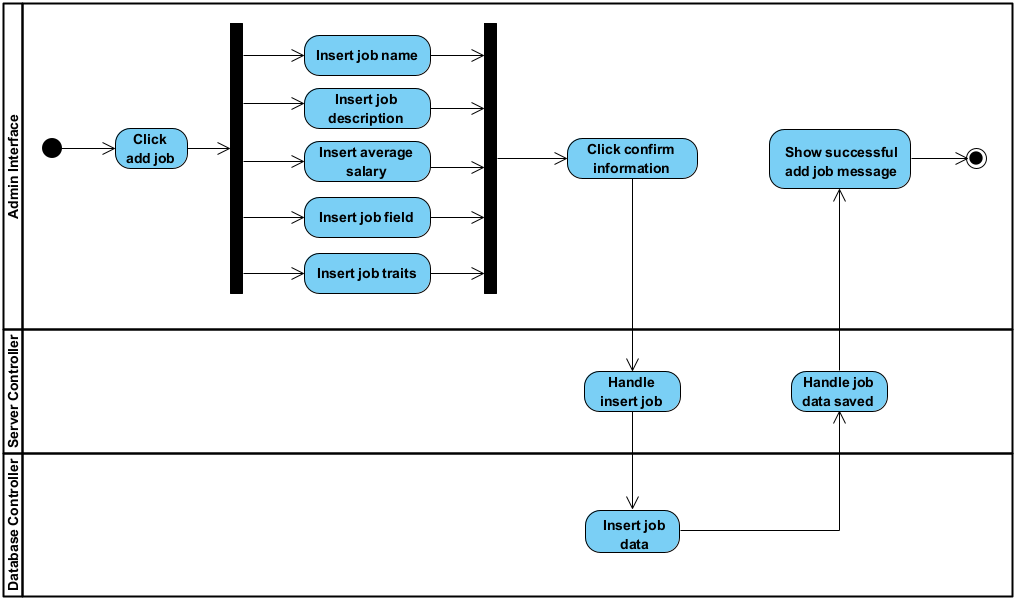
### 5.3.3. Activity Diagrams

#### 5.3.3.1. Searching job

**

*Figure9: Activity Diagram for searching job*

#### 5.3.3.2. Adding a new job to the database as an admin

*Figure10: Activity Diagram for add job by admin*

## 

## 5.4. Testing plan

| **Test** | **Module** | **Tested Function** | **Expected Result** | **Result** |
| --- | --- | --- | --- | --- |
| 1 | Registration | User Registration | Successful registration with valid user information. | **Pass** |
| 2 | Login | User Login | Successful login with registered credentials. | **Pass** |
| 3 | Login | Admin Login | Successful login with registered admin permissions. | **Pass** |
| 4 | Job Search | Search Jobs | Relevant jobs displayed based on the selected field. | **Pass** |
| 5 | Matching Algorithm | Trait Matching | Jobs recommended based on user traits using the genetic algorithm.The user is satisfied with the result. | **Pass** |
| 6 | Users Profile Editing | Edit Profile | Successfully update user profile information. | **Pass** |
| 7 | Admin Job Management | Add Job | Successfully add a new job listing to the system. | **Pass** |
| 8 | Admin Job Management | Delete Job | Successfully delete a job listing from the system. | **Pass** |
| 9 | Admin Job Management | Edit Job | Successfully edit an existing job listing in the system. | **Pass** |
| 10. | RAMAK Questionnaire | Answer Questionnaire | Allow users to answer the RAMAK questionnaire and store their responses accurately. | **Pass** |
| 11. | RAMAK Questionnaire | Answer Questionnaire | Can not submit until the user answers all the questions. | **Pass** |
| 12. | User's profile | Edit profile | Successfully update email | **Pass** |
| 13. | Admin Job Management | add permission | Successfully give editing permission on the site for a user from the list | **Pass** |
| 14. | RAMAK Questionnaire | RAMAK questionnaire scoring accuracy | Verify that the scoring logic for the RAMAK questionnaire works as expected and stores accurate results. | **Pass** |
| 15. | Login | Multi-device login functionality | Users can log in from multiple devices, and ensure session consistency. | **Pass** |
| 16 | Login | Login via gmail | Users can log in by gmail account | **Pass** |
| 16. | User profile | Ensure password reset functionality | Users can reset their password and that the reset link expires after a certain period. | **Pass** |
| 17. | User profile | Ensure user can replace profile image | Image profile of user changed | **Pass** |

*Table 1 - Table of test cases*

# 6. Expected achievements

The purpose of the site is to help users who are undecided about a job to achieve the job that suits them most.

This site will have several key themes.

* Data Organization: Systematically organize and structure the data obtained from the Facebook group to facilitate efficient processing and analysis.
* User Accessibility: Implement intuitive and user-friendly interfaces that enable seamless access to the website's data and functionality, ensuring a convenient experience for users.
* Jobs character traits: Analysis character traits required for jobs
* Administrative Control: Provide administrators with comprehensive control and oversight capabilities over the database of professions, user management, forum moderation, and other administrative functions.
* User Trait Evaluation: Employ state-of-the-art techniques and methodologies to accurately assess and analyze users' character traits, ensuring a precise understanding of their unique profiles.
* Genetic Algorithm: Optimal matching between character traits and potential careers

## 6.1. Project metrics

In order to quantify the success of our website and identify areas for improvement, we aim to implement a comprehensive evaluation system that will provide quantitative metrics to assess user satisfaction. This data-driven approach will enable us to refine the website's features and functionality, ensuring that it continues to meet the evolving needs of our users effectively.  
User satisfaction will be measured through several strategic methods:

* **Intermittent User Surveys:**

All the users will participate in surveys.

Surveys will use a rating scale (1-5) to gauge satisfaction with the algorithm's answers and the RAMAK questionnaire.

After collating a predetermined number of responses (150), user satisfaction scores will be analyzed.

The target is a high percentage (80%) of positive ratings (4 or 5).

Feedback will inform website refinements based on identified areas for improvement.

* **Genetic Algorithm Validation:**

Collaborate with a group of workers from the 'MITHABTEY' Facebook group who are satisfied with their current occupations.

The system compares the algorithm's recommended professions with the workers' actual fields of employment.

90% match of alignment between recommendations and occupations will indicate the algorithm's effectiveness.

* **Longitudinal follow-up:**

After 6 months we will contact some users to provide updates on their career progress.

This long-term feedback will enable assessing the lasting impact of the platform on users' career trajectories.

* **User Growth Targets:**   
  We anticipate a 30% increase in user registrations within the first three months of the website's launch. This metric will serve as an indicator of the platform's appeal and the effectiveness of our outreach efforts.
* **Surveys and Polls:**   
  We will organize surveys and polls within the 'MITHABTEY' Facebook group to gather information about users' experiences on the site. Our target is for at least 60% of respondents to recommend the site to others and at least 80% to express an intention to reuse the site when seeking career guidance in the future.

By implementing this comprehensive evaluation strategy, we will gain valuable insights into user satisfaction, the accuracy of our matching algorithms, and the overall impact of our platform on individuals' career journeys. This data-driven approach will inform our continuous improvement efforts, ensuring that our website remains a reliable and effective resource for career exploration and decision-making.

## 6.2. Challenges throughout the project

During the project development, we faced several challenges. One key issue was designing a scalable data model to handle relationships between users, jobs, and traits while ensuring fast data retrieval. We addressed this by structuring the MongoDB database with appropriate indexing and embedding related fields to reduce the need for complex join operations. Ensuring a seamless user experience across multiple devices, including handling responsive design and cross-browser compatibility, was another challenge. We resolved this by using Tailwind CSS for responsive layouts and conducting thorough testing on various devices.

We also faced challenges integrating third-party APIs, such as Google for social login, which required setting up OAuth 2.0 authentication. Using Passport.js simplified the management of Google tokens, allowing secure login. Another challenge was fine-tuning the genetic algorithm for job recommendations, ensuring accurate matches based on user traits. We used algorithm optimization techniques like memoization to improve performance. Finally, securing user data during registration and login involved implementing robust authentication with JWT tokens and bcrypt for password hashing, ensuring strong security throughout the system.

## 6.3. Results and conclusions

Researching a Questionnaire:  
We conducted extensive research to design a questionnaire that accurately captures a range of character traits. The questions were tailored to identify personal inclinations, strengths, and preferences in a manner that is scientifically grounded and easy for users to understand.

Developing the Genetic Algorithm:  
We created a genetic algorithm that matches individuals with the best-fit professions based on their character traits. The algorithm considers multiple factors, such as personality traits, preferences, and profession requirements, and iteratively optimizes the results to ensure the best possible matches.

System Adaptability:  
The system was built to be highly flexible and adaptive to changes in requirements, which were inevitable during the project’s development. Regular consultations with the client helped us adjust the features and ensure the system met evolving needs.

Character Trait Analysis for Professions:  
One of the biggest challenges we faced was accurately analyzing the relationships between character traits and professions. This task is complex and requires a deep understanding of psychology and occupational theory, typically the domain of experts in industrial psychology. We mitigated this challenge by talking with Ranan Hass.

Data Mapping and Validation:   
Ensuring that the algorithm consistently provided accurate matches required us to validate the data inputs and outputs at multiple stages. We coped with this challenge by testing extensively with various user profiles and adjusting the algorithm based on feedback.

Flexibility in Design:   
A key consideration in the project was building a system that could adapt to new requirements and changes. We designed the architecture to be modular, making it easier to implement new features or tweak existing ones without a complete overhaul.

## 6.4. Lessons learned

Effective Collaboration with Experts is Crucial

While we were able to develop a strong algorithm and questionnaire system, analyzing character traits for professions is a complex task requiring expertise in psychology and human behavior. Collaborating early with professionals from relevant fields could have enhanced the accuracy and depth of our trait-profession analysis.  
Future projects dealing with specialized domains should incorporate experts from the start, ensuring that complex topics are handled by those with the appropriate knowledge.

Flexibility in Design is Key to Success

The flexibility we built into the system allowed us to easily accommodate changes in requirements and evolving client needs. This proved essential in managing a project where new insights and adaptations were required throughout.  
Designing systems with modularity and flexibility from the outset ensures that changes can be managed with minimal disruption. This is particularly useful for projects with uncertain or evolving requirements.

Continuous Client Feedback Improves Project Alignment

Regular interaction with the client helped align the project more closely with their goals. By involving the client in feedback loops and testing, we were able to quickly adapt and make necessary changes.  
Open and continuous communication with stakeholders improves the likelihood of a successful outcome. Incorporating regular feedback cycles ensures that the project stays aligned with user needs and expectations.

Balancing Automation with Human Judgment

The genetic algorithm provided an effective method for automating the process of matching individuals to professions. However, automated systems may not always capture the full complexity of human psychology, highlighting the need for human judgment in final decision-making.  
Automation is powerful but has limitations. For projects dealing with human behavior or subjective elements, balancing algorithmic outputs with expert review or human judgment can improve the quality of results.

User-Centric Design Enhances Usability

Focusing on user experience (UX) during the design of the questionnaire and interface ensured that the system was easy to use, which directly impacted user engagement and satisfaction.  
Prioritizing UX in the development process leads to a more intuitive and accessible product, increasing overall user engagement. Usability should be considered at all stages of design.

## 6.5. Compliance with setting goals

At the outset of the project, we established a range of ambitious goals to ensure the system's functionality and user experience. One of the key objectives was the systematic organization of data received from the Facebook group, which we successfully achieved. By organizing the data efficiently, we facilitated smoother processing and analysis, ensuring the system operated seamlessly. Another critical goal was to ensure user accessibility by designing intuitive and user-friendly interfaces. We accomplished this, creating a system that provided users with a comfortable and straightforward experience when accessing the website's data and functionality. Additionally, our goal to give administrators comprehensive control over the database, user management, and forum functionalities was fully realized, ensuring robust administrative oversight. Finally, we implemented advanced methodologies for accurately assessing and analyzing users' character traits, meeting our objective to provide an in-depth understanding of user profiles.

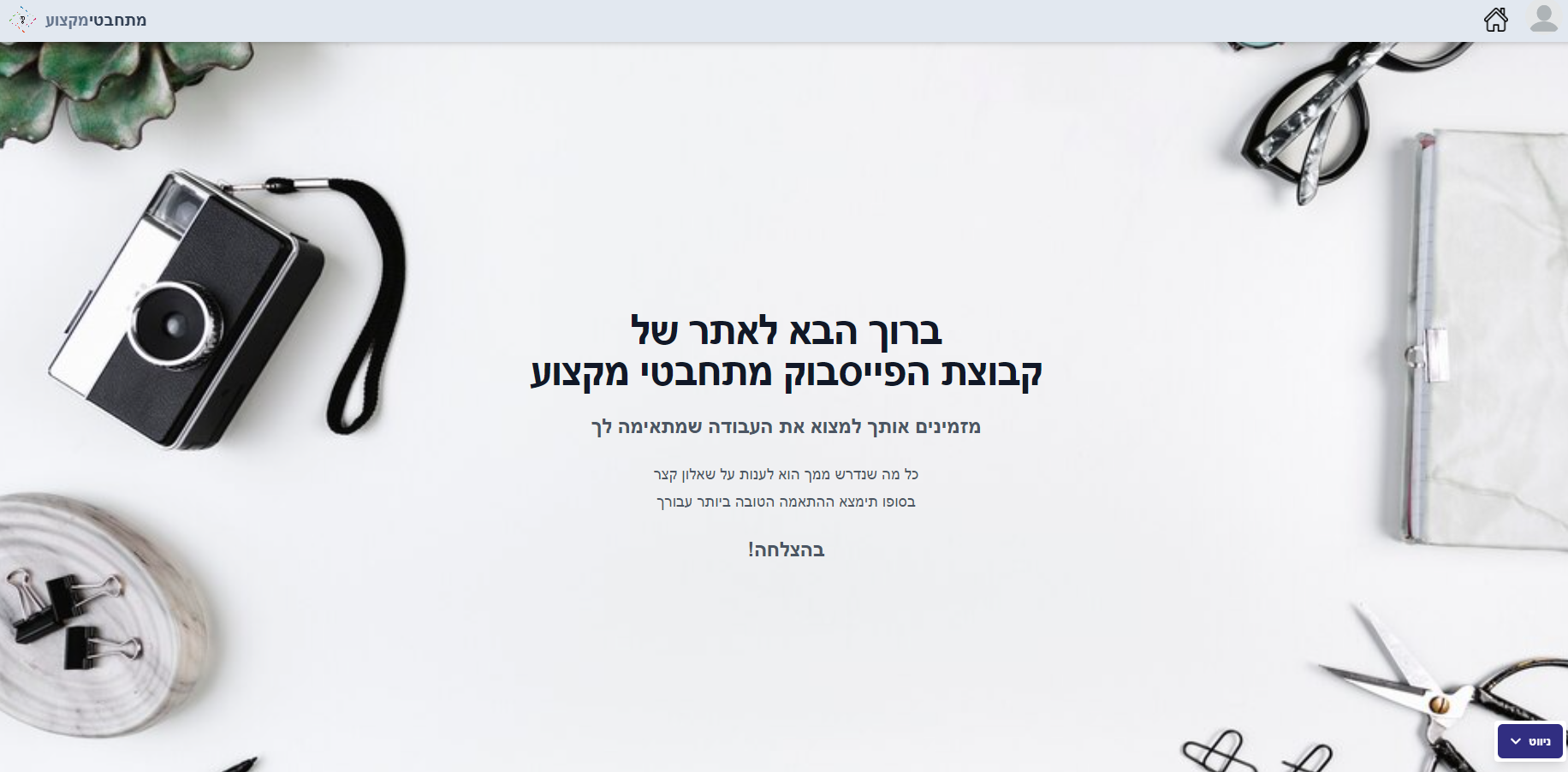
However, despite these successes, some challenges emerged. Our goal of analyzing the character traits required for different jobs was only partially met. The complexity of this task, which involved aligning character traits with job requirements, required more time and expertise in occupational psychology than initially anticipated.

While we made progress, further refinement and collaboration with industry experts would be necessary to fully achieve this goal. Nevertheless, our efforts to implement a genetic algorithm for matching users' traits with potential careers were successful. The algorithm optimized the matching process, providing users with personalized career recommendations based on their unique traits, meeting one of the project's most important goals.

# 7. User Guide

## 7.1. Home Page

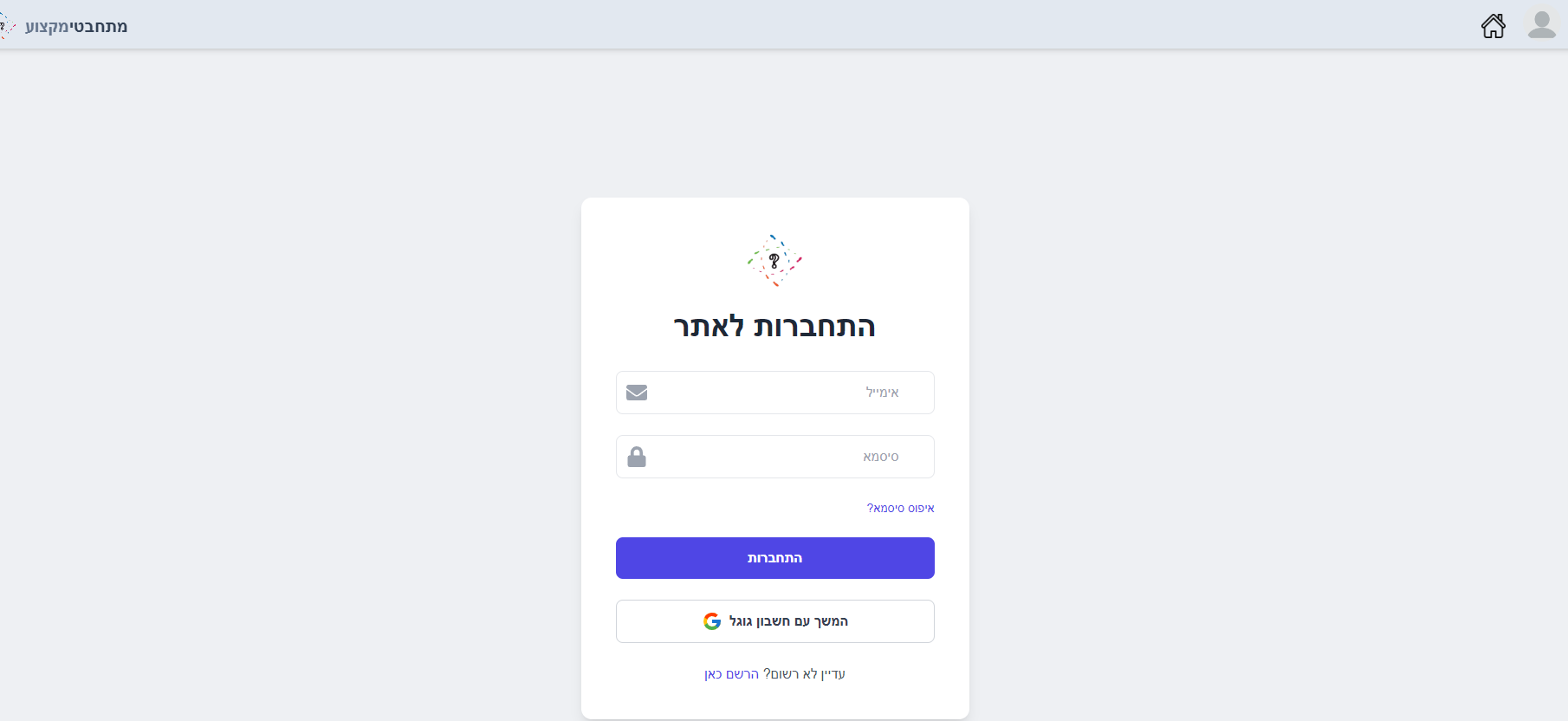
This page serves as the landing page for your site, which is part of the project for the Facebook group "מתחבטי מקצוע." From this page the user can navigate to all pages in the project. In this page, the user also can read about the Facebook group and its manager.



*Figure11: Home Page*

## 7.2. Login Page

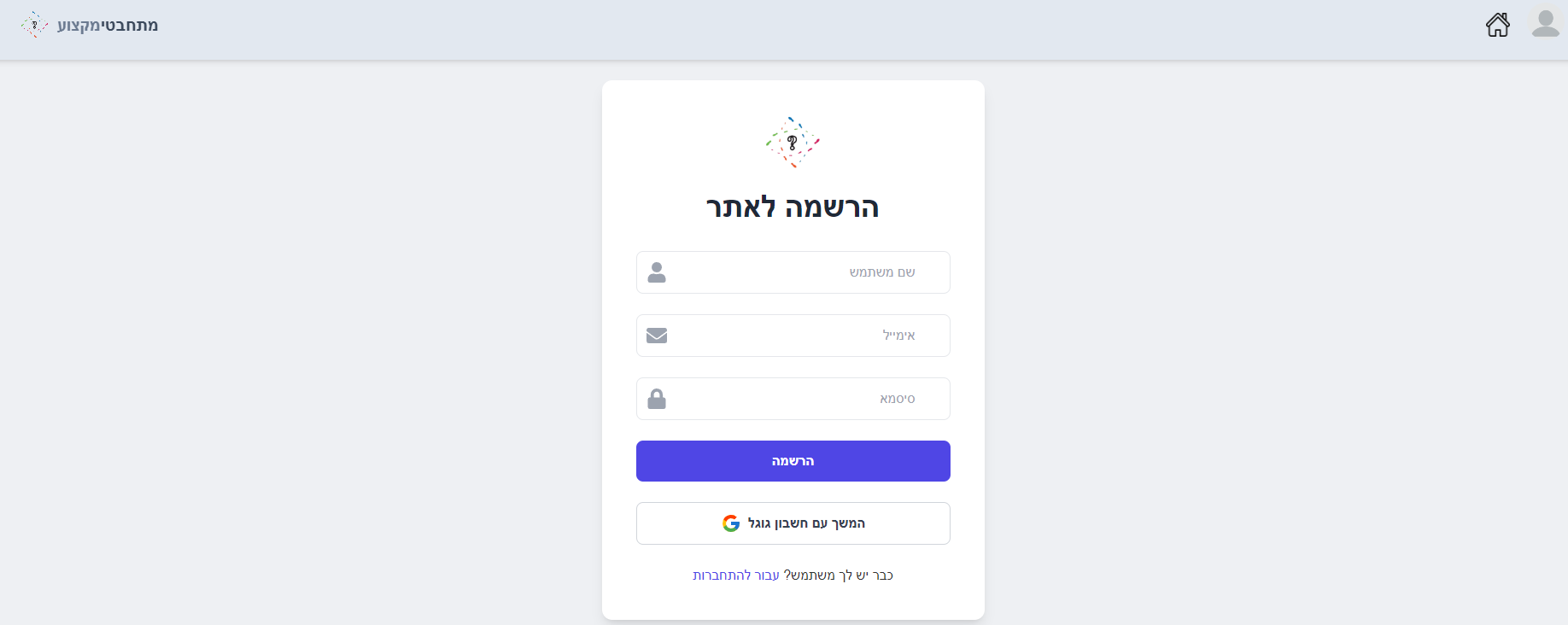
## 7.3. In this page the users can login to the website or register as a new user.

The user can register by email and password or by gmail account.

*Figure12: Login Page*

## 7.4. Register Page

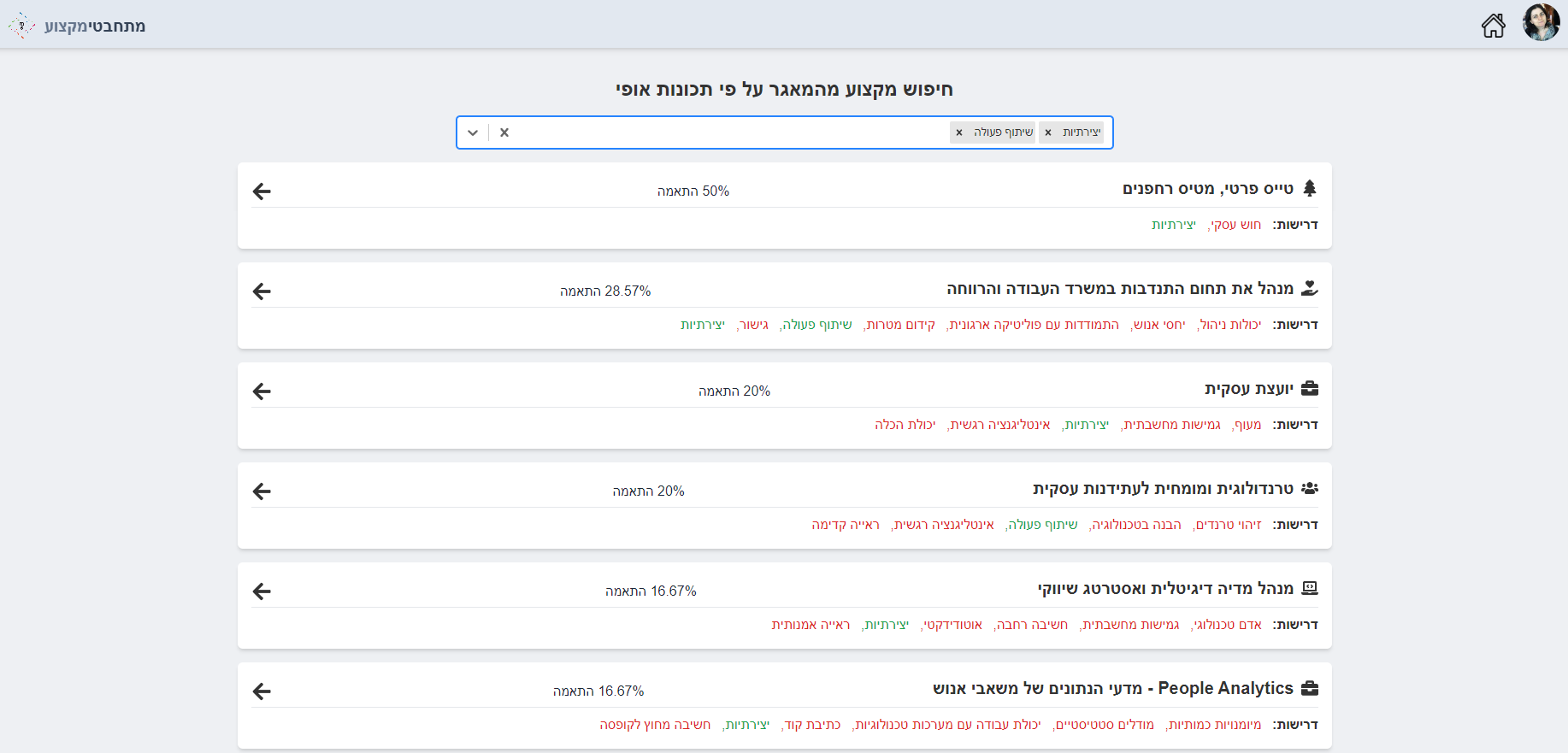
In this page the users can register to the website by inserting a unique email, password and username.



*Figure13: Register Page*

## 7.5. Searching Job By Traits Page

In this page users can explore job opportunities by inserting traits to filter them, and see the percentage of matching between traits that the user inserted to the job`s traits. In addition, the user can view the job details.

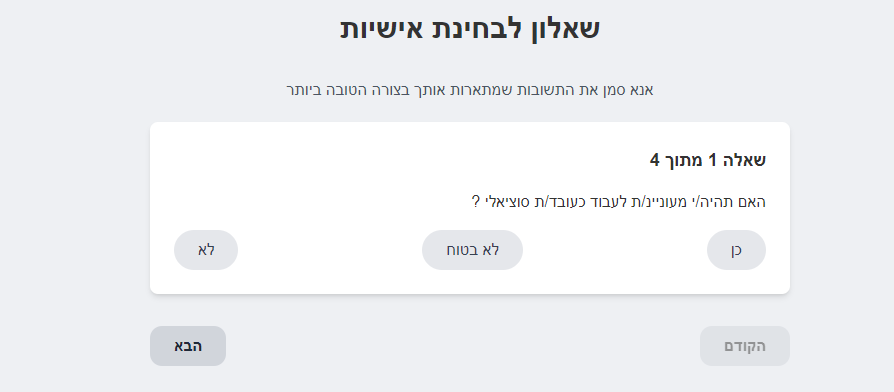


*Figure14: Page for searching job by traits*

## 7.6. RAMAK questionnaire Page

In this page the user fills in his answers to the questionnaire.

After he finishes answering all the questions, a calculation is made that describes his traits and then the 3 jobs that best suit him.

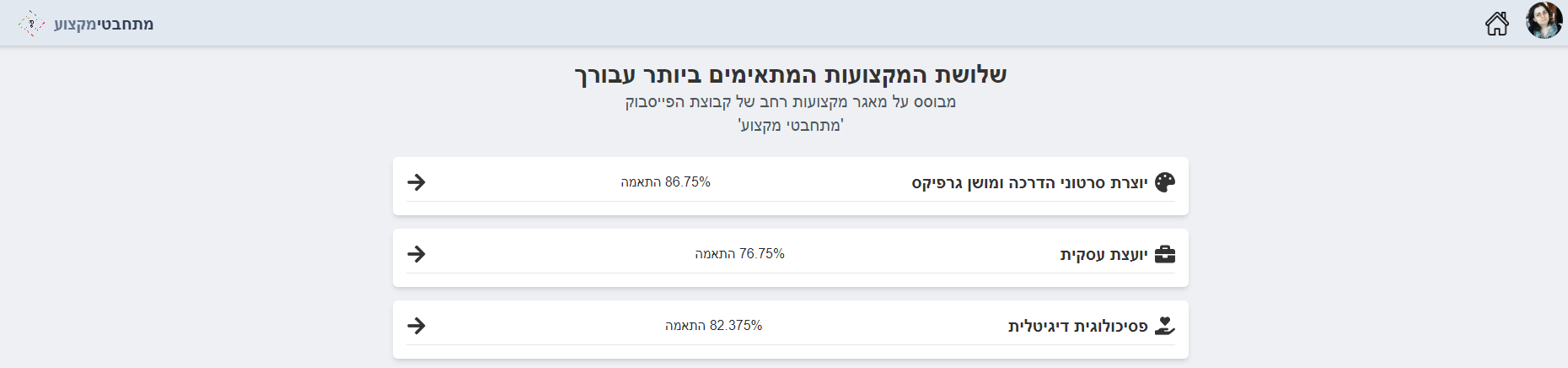


*Figure15: Ramak questionnaire Page*

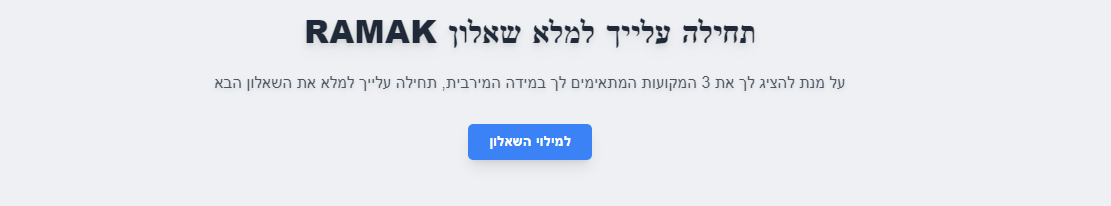
## 

## 7.7. The most 3 suit jobs Page

In this page the users view the 3 jobs that most suit them, after they fill the RAMAK questionnaire.

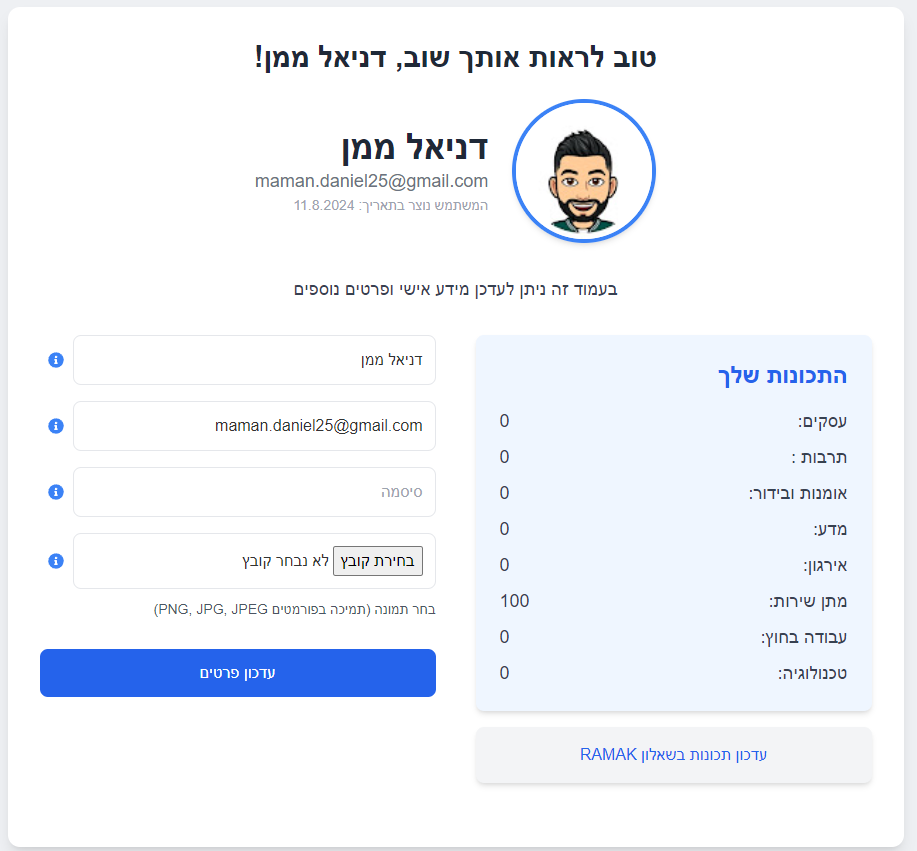


*Figure16: Three most suit jobs page*

If the user did not fill the RAMAK questionnaire yet, it asks him/her to do it.  
  


*Figure17: Start ramak questionnaire Page*

## 7.8. User`s Profile Page

In this page the users can update their data such as: username, email, password or their image and see their RAMAK questionnaire if there is.

*Figure18: User`s profile Page*

## 

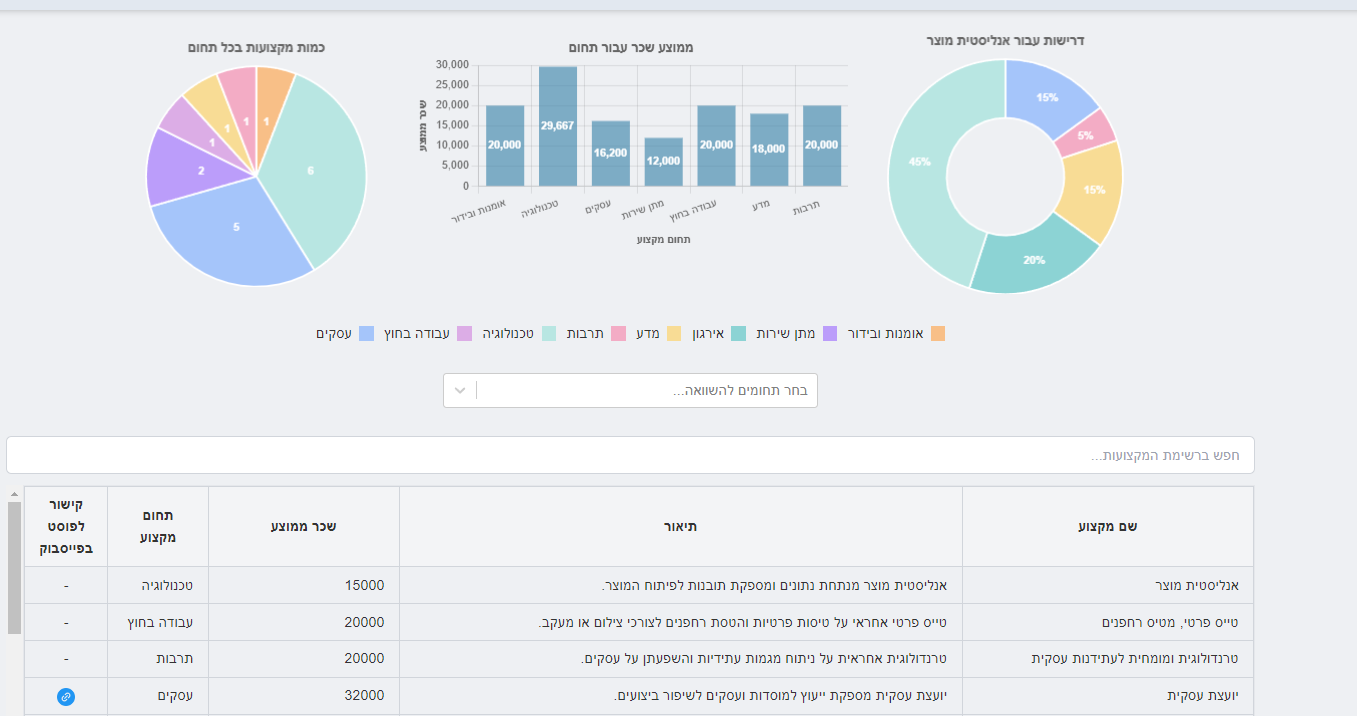
## 

## 

## 

## 7.9. Information of jobs that are characterized by RAMAK questionnaire Page

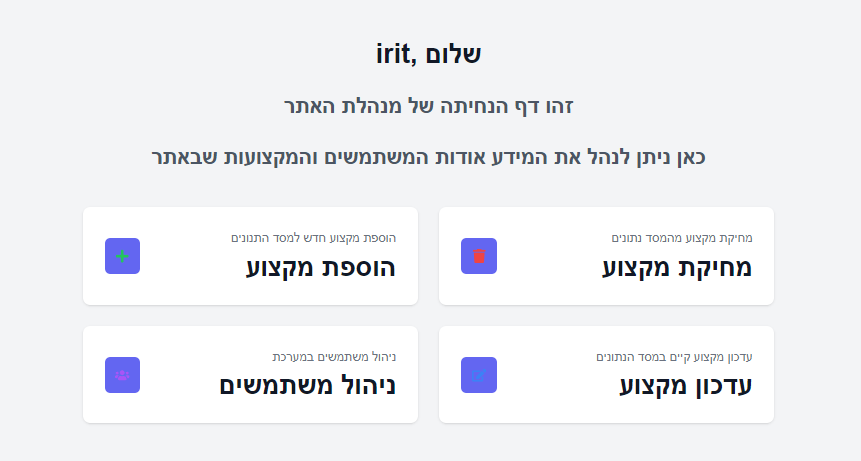
In this page the user can view information about professions that exist in the system for the RAMAK questionnaire. The user can make a comparison between the different fields that this questionnaire has and see details for a specific job.



*Figure19: Page that give information of jobs*

## 7.10. Admin Panel Page

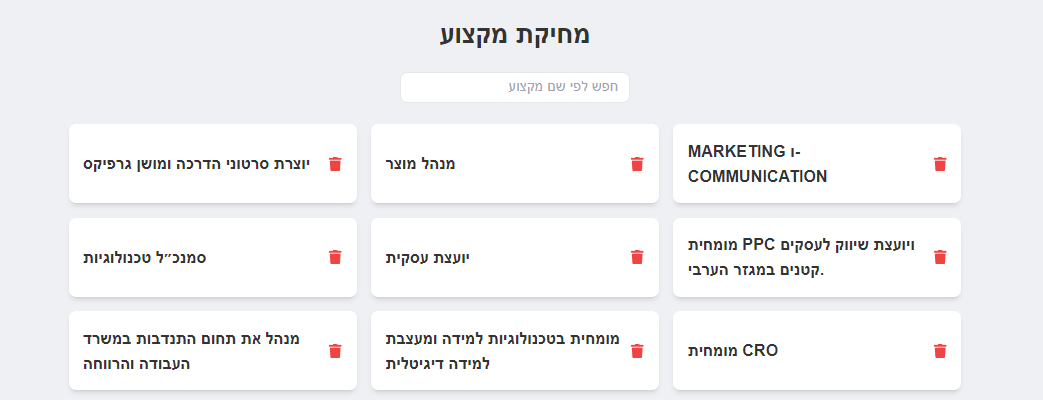
Admin landing page. In this page the admin can choose whether to navigate to add a job, delete a job, manage users permission or update a profession



*Figure20: Admin Panel Page*

### 7.10.1. Delete Job Page

In this page, the administrator will be able to view the existing jobs, search for a specific job by name and delete a job from the database.

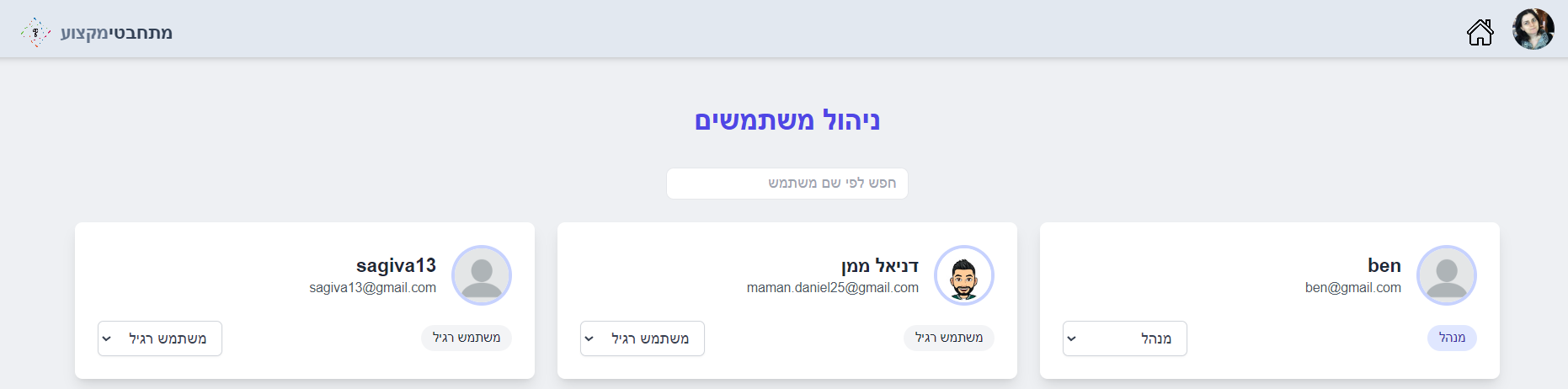


*Figure21: Admin Panel - Delete Job Page*

### 

### 7.10.2. Users Permissions Page

On this page, the main administrator (Irit only) can update user permissions.  
The permissions for the users are for the benefit of providing help to the main administrator in order to add / delete / edit professions



*Figure22: Admin Panel - Users Permissions Page*

### 

### 

### 

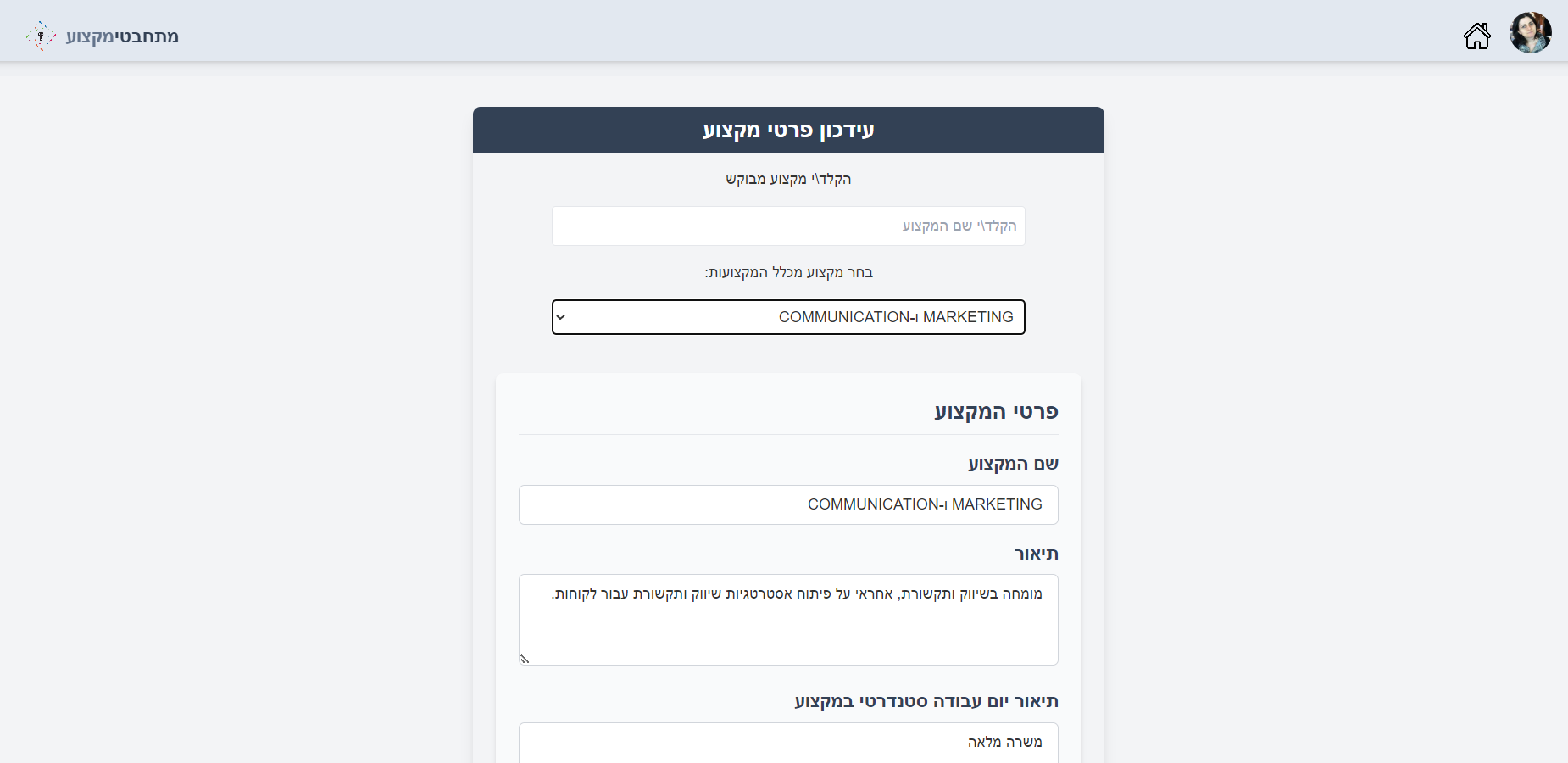
### 

### 

### 7.10.3. Update Job Page

In this page, the administrator can edit the details of an existing job.

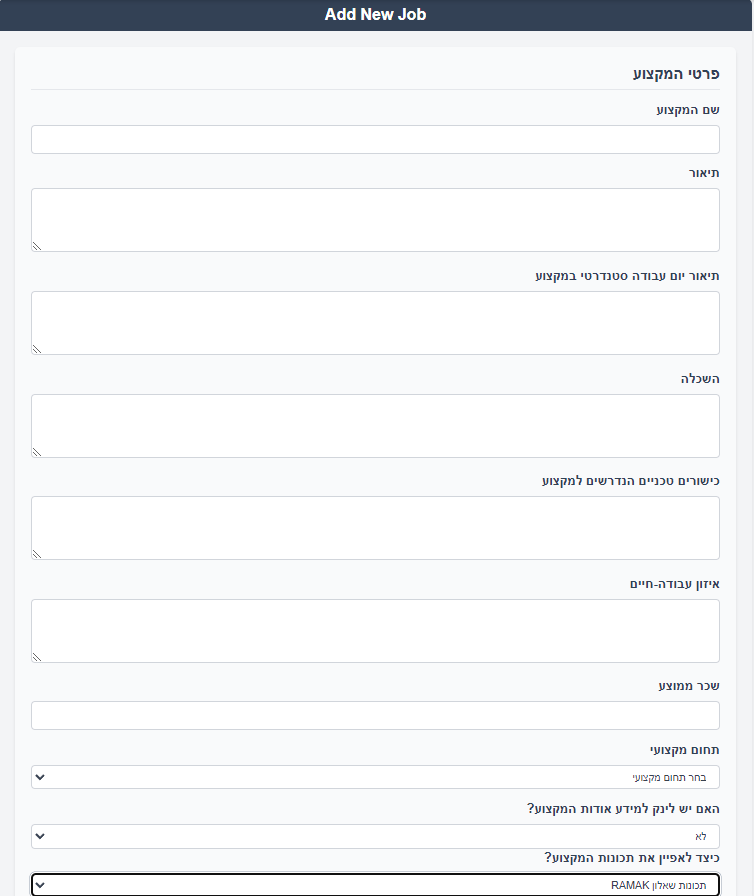
When a job is selected, a form will open with all the details of the job.



*Figure23: Admin Panel - Update Job Page*

## 

### 7.10.4. Add Job Page

On this page, the administrator can add a job to the database. The admin will fill in all the required fields such as: name, description, averageSalary, field, prerequisites, url to post on the job, general requirements, standard day, education needed, technical skills, work life balance. Admin can describe the job by RAMAK traits or general traits.

*Figure24: Admin Panel - Add Job Page*

# **8. Maintenance Guide**

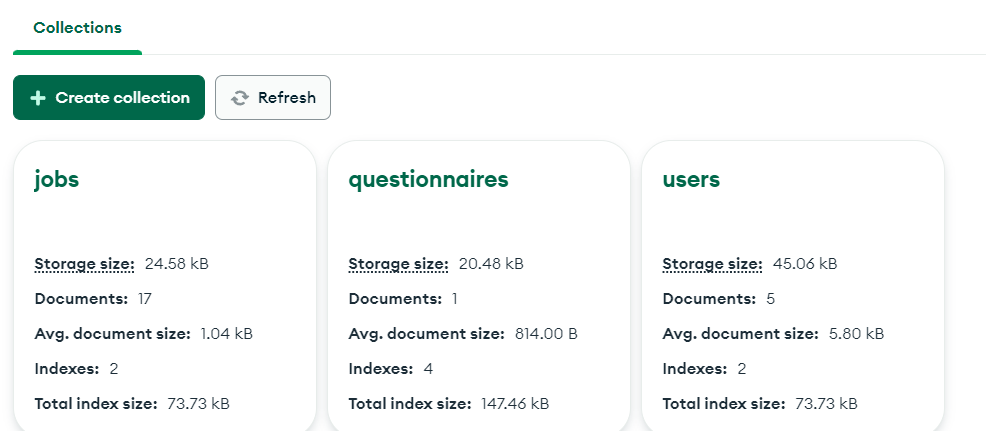
## **8.1. General Description**

The purpose of the system is to facilitate the process of job matching and management on your web platform. The system consolidates all necessary processes, making job searching and management easier for users, including job seekers, and administrators. The job matching process is enhanced by the use of the RAMAK questionnaire and a genetic algorithm, ensuring accurate and personalized job recommendations.

Today, many tasks like job searches, profile management, and job postings are scattered across various platforms. Our system unifies these tasks into one streamlined web application. It efficiently matches job seekers with suitable roles by analyzing their character traits and provides tools for managing job listings and user profiles.

## 8.2. Database

The main documents in our system are the User, Job, and Questionnaire documents. The collection of users is designed to store the users and their details on the site. The collection of jobs is designed to save the details of all the jobs and their characterization. In addition, the collection of the questionnaire is designed to save the questions of a questionnaire.

****

*Figure25: Database collections*

## 

## 8.3. Usage Scenarios

### 8.3.1. Job Seeker

1. Registers on the platform.
2. Login via username and password.
3. Login via gmail.
4. Completes the RAMAK questionnaire to assess personality traits.
5. Receives personalized job recommendations.
6. Edits and updates their profile.

### 8.3.2. Admin

1. Authorizes permissions for employers.
2. Manages user accounts.
3. Manages job listings (add, edit, delete).
4. Oversees the RAMAK questionnaire results.
5. Runs the genetic algorithm to match job seekers with suitable traits.

### 8.3.3. Software Environment

1. Windows operating system
2. MongoDB, Express.js, React, Node.js (MERN stack)
3. Visual Studio Code
4. Node.js

### 8.3.4. Running Instructions

1. Set up the MongoDB database by running the necessary scripts.
2. Install required packages:

- npm i tw-elements

- npm i jsonwebtoken

- npm i bcryptjs

- npm i redux-persist

- npm i cookie-parser

- npm i chartjs-plugin-datalabels

- npm i firebase

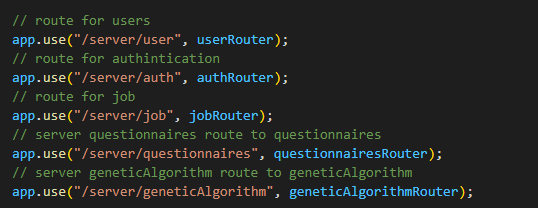
- npm i react-select

- npm i @heroicons/react

1. Start the backend server by navigating to the server directory and running npm start.
2. Start the frontend client by navigating to the client directory and running npm start.
3. Access the web application by navigating to http://localhost:5173/ in a web browser.

### 8.3.5. Routing

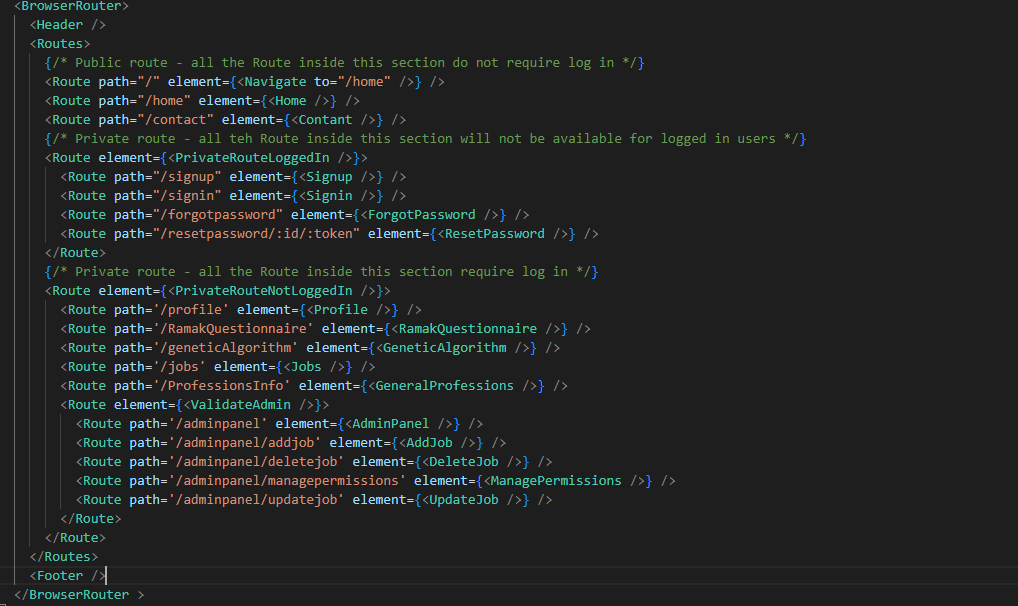
The application defines routes for different functionalities.



*Figure26: Routers exist in server*

* /server/user for user-related operations (handled by userRouter)
* /server/auth for authentication-related operations (handled by authRouter)
* /server/job for job-related operations (handled by jobRouter)
* /server/questionnaires for questionnaires (handled by questionnairesRouter)
* /server/geneticAlgorithm for genetic algorithm-related operations (handled by geneticAlgorithmRouter)

### 8.3.6. Security

*****Figure27: Permissions of pages for clients*

* **Routing Structure with BrowserRouter:**  
  The BrowserRouter component from react-router-dom wraps the entire app, enabling client-side routing. All routes defined within Routes are accessible via browser navigation without reloading the page.
* **Public Routes:**  
  The first set of routes is public, meaning no login is required to access these pages.
* **Private Routes (For Logged-in Users):**

The second set of routes, wrapped inside <PrivateRouteLoggedIn />, ensures that these pages are only accessible to users **who are not logged in**.

* **Private Routes (For Logged-out Users):**

The third section, wrapped inside <PrivateRouteNotLoggedIn />, restricts access to users **who are logged in**.

* **Admin Routes:**

Routes under <ValidateAdmin /> are only accessible to **admin users**.

# 9. References

[1] Katoch, Sourabh, Sumit Singh Chauhan, and Vijay Kumar. "A review on genetic algorithm: past, present, and future." Multimedia tools and applications 80 (2021): 8091-8126.‏

[2] Kern, M. L., McCarthy, P. X., Chakrabarty, D., & Rizoiu, M. A. (2019). Social media-predicted personality traits and values can help match people to their ideal jobs. Proceedings of the National Academy of Sciences, 116(52), 26459-26464.

[3] Mathew, Tom V. "Genetic algorithm." Report submitted at IIT Bombay (2012): 53.‏

[4] Meir, Elchanan I., et al. "Examination of interest inventories based on Roe's classification." The Career Development Quarterly 46.1 (1997): 48-61.

[5] nodeJs: <https://nodejs.org/en>

[6] react: <https://react.dev/>

[7] jwt: <https://jwt.io/>

[8] render: <https://render.com/>

[9] mongoDB: <https://www.mongodb.com/>

[10] tailwind: <https://tailwindui.com/>

[11] TypeScript: <https://www.typescriptlang.org/>

[12] 'Mithabtey Miktzoa' facebook group:<https://www.facebook.com/groups/612789808874188>

[13] MERN architucture : <https://medium.com/techiepedia/what-exactly-a-mern-stack-is-60c304bffbe4>

[14] GA Roulette wheel selection: <http://www.edc.ncl.ac.uk/highlight/rhjanuary2007g02.php>

# 10. Appendix

## 10.1. Questions for interviews in the Facebook group

* What are the required or optimal qualities to work in the field?
* What does a standard working day look like in the field?
* How do you get hired to the field? How can you join it?
* Do you need a degree? If yes, in which field?
* Are technical skills necessary to work in the field?
* What experience is needed to get a job in the field? (if there is one)
* How is the work-life balance in the field?
* When and how did you know you arrived at the job that is right for you?
* What is the salary in the field?
* What is the less glamorous side of the job?
* Are there any unique challenges for this field?
* Are there any trends in the field?
* Will the profession disappear with technological advancement?
* Is there a sub-field in your field that is going to overcome it in a decade or two?
* Will the technological advance or the changes in the field of information gathering and AI also affect the field?
* Do you think there will be changes in the criteria for defining the position in the field in the coming decades?
* Is there a close field that may merge with the field?
* Do you see a change in the way they will work in your field in the coming years?
* How do we know how to identify a job in the field?
* Is there a close field that you think will disappear over time due to technological advancement?
* What do you like most about this job? What is it magic?

## 

## 10.2. Meeting schedule with client - Irit Hommasi

* 13.9.23 - Introductory conversation and formation of an idea for a product
* 17.1.24 - Meeting after the return of studies
* 31.1.24 - Meeting to present information retrieved from Facebook
* 14.3.24 - Meeting for website development and characterization with a psychological expert
* 18.4.24 - Status meeting - Introduce admin panel
* 08.5.24 - Status meeting - Implement genetic algorithm
* 10.7.24 - Status meeting
* 26.8.24 - Status meeting - Introduce searching by simple traits
* 15.9.24 - Final product approval meeting