

Quick Match

A future choices supporting platform



Our team

*Platform was developed with
passion and commitment by :*



- **Michał Grochowski**
- **Jakub Skrzyński**
- **Adam Sieczka**
- **Wojciech Boduch**
- **Marcin Niemczyk**
- **Adam Nieckarz**

Main goal of the system

Based on the gathered and preprocessed data, system is capable of helping user to decide which school/university/field of study is suiting he/she the most. User is being asked questions about importance of the parameters of schools included in system in a form of a quiz and then the ranking of most suiting options is presented.

Idea of system - multicriteria optimization

- Criterias are defined as questions
- Utility function is defined for each school/field of study as a sum over all possible parameters of $parameter_value[school/field]*weight[parameter]$
- We try to maximize function value - based on config we provide n best results.
- System selects questions that will split the set of possible alternatives the most.
- User may rate the criteria - set its weight - based on enum (hate the parameter=-2; don't like=-1; no opinion=0; like=1; love=2)
- Parameters are defined as integer numbers. Negative value means opposition to parameter and positive value means presence of the parameter. The higher the absolute value the stronger the relation is.

Parameters and questions

- Questions do cover important parameters of school, like location, discipline, rules, additional important factors like character of city, earned titles and earned qualifications.
- Values of parameters should be designed to show adjacency or opposition of the given school with the parameter with positive or negative values respectively - the higher absolute value the stronger relation. For example in case of question “How would you rate location of school in the south of country”, universities like AGH should get high positive value and University of Gdańsk should have value that is near to negative value of the AGH’s value.
- Questions should cover all important features of presented options:
Location, surrounding, ease of commuting, job opportunities, social life, costs, organised events, ease of enrolment, requirements, average amount of effort and so on.

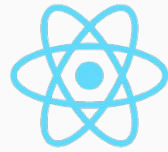
Source of data

- Important is to prepare questions that will describe well the selected schools. They are not included in system setup - it is considered as configuration data stored in sqlite.
- Source of parameter values should be facts (like location, average costs of living ...) and surveys conducted among students of the given university (amount of social events, ease of studying, atmosphere ...)
- System in demo version is populated with random data.

Tuning of the system

Tuning can be done by adjusting two parameters - desired amount of options to be displayed and minimal relative difference between best and first rejected option. These parameters are also considered to be configuration data in demo stage as they are very dependent on gathered data set.

Used technology



React



Tailwind CSS

SQLAlchemy

NEXT.js



SQLite

AXIOS



{JSON}



Advantages

- Highly customizable - parameters values, questions and possible choices are to be set in the database as well as parameters.
- System tries to limit amount of asked questions based on quick heuristics.
- System is platform independent and split into two parts - front web gui and rest api server
- Scores assigned in ranking are mathematically explainable, which would not be the case in neural network based solution.

Deployment

Application requires very little resources and provides multi platform support. Platform was designed to state as little requirements as possible. It requires a python supporting server to host a fast api backend and a web server capable of supporting react frontend. Both of them does limit memory and computational complexity.