PROGRAMMING
University of Victoria Engineering Competition

Jayden Chan jaydencn7@gmail.com
Cobey Hollier cobey.hollier@mac.com

September 26 2020

1 Situation

Your group has been tasked with creating an online dating platform. You are responsible for creating a visual component for showing potential matches, along with an algorithm for matching users together. Each user has attributes that can be used to determine suitable matches such as location and age, (see **section 2.1** for a full list of requirements). You will be provided with data in several common formats that will serve as your pool of users.

2 Deliverables

Submit your deliverables as a ZIP file emailed to the organizers by 5pm PDT, Sept 26 2020.

- Source code (**section 2.1**)
- Presentation (**section 2.3**)
- Documentation (**section 2.4**)
- Instructions for how to build and run your project on both Windows **and** MacOS/Linux systems
- Link to version control repository (GitHub/GitLab etc)

2.1 Software Requirements

There are three levels of functionality that you should try to implement. Each level will build on the previous level and add to your overall score. The purpose of these levels is to provide concrete goals to work towards and to change the scope of the project based on what you think you can accomplish in the allotted time. You do not need to complete the objectives in order, but doing so is recommended.

2.1.1 Program Logic

To begin, create a program that can accept two user's emails and output a score that indicates how good a match those two people are. You may use the provided user attributes any way you want to create your matching algorithm. For this level, your program can run on the command line as the emphasis is on the algorithm instead of the usability. Your program should be capable of generating a score for any pair of users and should output a score from 0 to 100.

In addition to showing the score for two individual users, you should be able to provide a single user's email and get a sorted list of their top matches. The scores should be determined using the same algorithm that you created in step 1.

The user data will be provided in JSONL (line separated json), CSV and TSV (tab separated) format for your convenience.

2.2 Application Program Interface

For the second level of this project you will need to create a web API for your frontend to use in level 3. This could be something like a REST server, but you are permitted to use any protocol you want including GraphQL, gRPC, SOAP, Websocket etc.

2.2.1 User Interface

In order to increase the projects accessibility and aesthetics, a frontend can be built that will display your output from level 1. The scope of the frontend in up to you, but you must at least show the sorted list of "top matches" for a user. If you have additional time, you may add any additional features or information you want to your frontend. This is a good area to increase your points in the "creativity and innovation" section.

2.3 Presentation Requirements

Presentations will be conducted online via Discord after the scheduled programming time. Your team must create a 10-15 minute presentation that includes a demonstration of your code. It is recommended to create some form of visual accompaniment such as a slide show.

Your presentation could include (but is not limited to):

- Design overview
- Rationale for design choices
- 3rd party libraries & rationale
- Reported bugs

- UI screenshots

2.4 Documentation Requirements

In addition to your code and installation instructions, you should create a brief report to explain some decisions that are not shown through code alone.

This report could include (but is not limited to):

- Information about your design decisions
- The division of labour among your teammates
- Any open-source libraries used, and why you chose them
- Ways that you would expand your product if you had more time

3 Rules

Only original projects, which are the work of the competitors, may be entered. An original project is one that has not been entered in the University of Victoria Engineering Competition in previous years. The project must not have a common or obvious approach, solution, or application. This means that approaches that copy significant amounts of code from a similar project online will be disqualified. Use common sense and treat this competition with the same academic integrity rules as you would any other UVic Software Engineering assignment.

You are permitted to use any free and open-source libraries in your project.

You are permitted to use cloud-based compute services such as AWS/GCP/Heroku/Azure/DigitalOcean etc.

It is up to the competition organizers' discretion to determine when a rules violation should result in a reduction of score or disqualification.

4 Judging

Judging will begin after the programming period has ended at 5pm PDT. The presentation order will be randomly generated and announced at 5pm so that teams with later presentation times cannot plan for extra preparation time. Judges and contestants will join the discord waiting room at their prescribed time. You will be admitted for your presentation once the previous group is done. Please wait in the waiting room until you are admitted.

5 Judging Rubric

Description	Points
Technical	**/70**
Solution Feature Functionality	/30
Program design quality	/15
Documentation	/15
Innovation and creativity	/10
Presentation	**/30**
Organization	/10
Confidence and presence	/5
Communication	/5
Visual aides	/5
Responses to questions	/5
Penalties	
Reported bugs/issues	No penalty
Non-reported bugs/issues	-20 each
Plagiarism	Disqualification
Total	**/100**

The scores assigned by judges are final and cannot be negotiated after the winners are announced.