PROJECT 1

PROJECT NAME	**Emerging geo-data sources to reveal human mobility dynamics.**	CLIENT	Bruce Sham (b.sham@auckland.ac.nz)
PROJECT DELIVERABLE	1) documenting the algorithm as a user manual; 2) making the algorithm available for others to execute in R/Python (e.g. on GitHub); and 3) the resultant shape file (.shp) with the foot traffic information that can be opened in ArcGIS; a resultant file is attached.		
PREFERRED SKILLS	C programming		

BACKGROUND

Effectively monitoring the dynamics of human mobility is of great importance in retailing strategy.

Traditionally, human mobility data is collected by roadside sensors, which have limited spatial coverage and are insufficient in large-scale studies.

With the maturing of mobile sensing technologies, various crowdsourced data are emerging, paving the way for monitoring and characterizing human mobility for trade area analysis, site selection, competitive intelligence, or consumer insights.

PROJECT DESCRIPTION

This project aims to emerge three mobility data sources, including mobile device data, point of interest (POI) data, and the Council's geospatial data and provides the foot traffic insight in New Zealand (NB Auckland as a protocol in this study).

PROJECT 2

PROJECT NAME	**Wearable and Mobile Enabled Smart Aged Care.**	CLIENT	Jing Sun jing.sun@auckland.ac.nz
PROJECT DELIVERABLE	A mobile application together with wearable devices to support aged care.		
PREFERRED SKILLS	AWS, Flutter, .Net		

BACKGROUND

Wearable technology has become increasingly integrated into our daily lives, which gathers useful information, e.g., health and exercise data of their wearers. Currently, common fitness wearables provide real-time statistics as well as a history of exercising metrics. There are applications that analyze the collected data and report back to the users on their achievements. However, most existing applications only perform passive data collection rather than acting as a proactive advisor that improves the physical status and provide expert feedback and monitoring to the user.

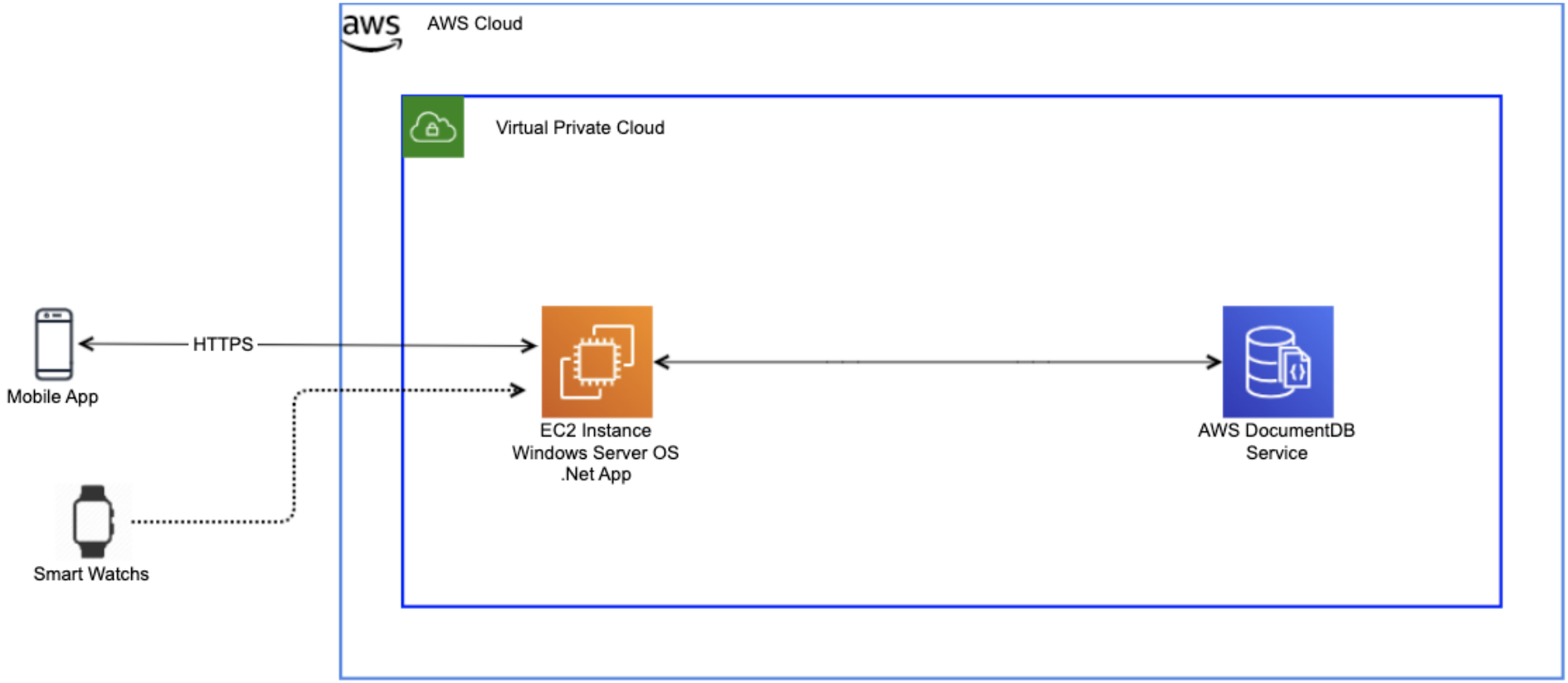
PROJECT DESCRIPTION

This project aims at providing an artificial intelligence solution backed by research in the aged care domain. The outcome will be an "AI Personal Carer" that provides the user with both real-time and long-term user advice, such as daily routine monitoring, reminder/caring, as well as risk alert.

Project's Research Components:
1) Research and understand the data collected by the wearable device and how it can be used in the context of aged care.
2) Investigate AI/Machine Learning techniques that can utilize research in aged care and provide a viable solution.

Project's Implementation Components:
3) A mobile and Internet based software application that realizes the proposed solution.

The proposed system architecture is as follows:



CONSTRAINTS

The software implementation will be owned by the project owner for future development and extension purpose.

PROJECT 3

PROJECT NAME	**StudyPal**	CLIENT	Allan Fowler allan.fowler@auckland.ac.nz
PROJECT DELIVERABLE	An app that can enable students to search for study pals based on year level, subject, gender, and location.		
PREFERRED SKILLS	Java, App development, UI Development, UX testing		

BACKGROUND

StudyPal is a mobile app to help students find someone to study with.

PROJECT DESCRIPTION

The app would need to enable students to search for study pals based on year level, subject, gender, and location (MVP). Other search criteria are welcome.
The MVP would be for Android KitKat (and above). Alternatively, a beta version for iOS (> 15.0) would be considered.

AVAILABLE RESOURCES

Training materials, web resources.

Students would need to have access to a cell phone for testing

PROJECT 4

PROJECT NAME	**Inner PEACE App**	CLIENT	Allan Fowler allan.fowler@auckland.ac.nz
PROJECT DELIVERABLE	At a minimum, the app must include at least two activities that can help students self-manage their anxiety or stress. The app must be able to run on Android KitKat or iOS 11. The MVP should be a working polished beta..		
PREFERRED SKILLS	Java, App development, UI Design, UX testing		

BACKGROUND

Stress and anxiety are common experiences among students, particularly in high-pressure academic environments such as colleges and universities. Many factors can contribute to student stress and anxiety, including academic workload, financial pressures, social pressures, and personal issues.

PROJECT DESCRIPTION

The inner PEACE App will help students self-manage anxiety and stress through activities like breathing, light exercise, meditation, listening to music, or playing a game. The App will support users through interactive guidance.

AVAILABLE RESOURCES

Graphics, Concept art, etc.

CONSTRAINTS

Students should not release the app into the wild (Google Play or iTunes Store). We will undertake a research study before release.

PROJECT 5

PROJECT NAME	**The Eye of Auckland**	CLIENTS	Jing Sun jing.sun@auckland.ac.nz
PROJECT DELIVERABLE	A mobile application that can remind the general population to keep away from dangerous incidents.		
PREFERRED SKILLS	Mobile development technology (e.g., Android), web crawler technology, some elementary knowledge about data mining, and API on Google Map, Twitter, Facebook, etc.		

BACKGROUND

Nowadays, there are many insecure factors happening in our society, such as natural disasters (hurricane, rainstorm, earthquake, etc.), crime related incidents (burglary, assault, etc.), which cause serious consequences to our personal daily activities.

PROJECT DESCRIPTION

This project aims at developing a mobile App (via Android development platform and kits) to report insecure information within a city environment. A brief project requirement is described as follows.

1) Data collection: web crawl social media sites on the emergency events or incidents, such as historical datasets on Twitter, Facebook, and other social network websites. These historical datasets may include user profile, timestamp, location (Longitude and latitude, etc.) and so on.

2) Data processing: (a) unify the collected data into standard datasets, and calculate the frequency of events in a chronological order. (b) provide definitions to specify insecure factors. For instance, what represent an insecure factor? What insecure factors exist in our city? Whether certain metrics can be proposed to compute these insecure factors? Please present simulation models and algorithms based on your idea.

3) Data visualization: display the end results on Google map via data visualization methods, including but not limited to heat map, scatterplots, etc.

AVAILABLE RESOURCES

1.https://developer.android.com/ 2.https://developers.facebook.com/docs/android/ 3.https://developer.twitter.com/en/docs/twitter-api
CONSTRAINTS
Better for coding web crawl, auxiliary tool also permitted.

PROJECT 6

PROJECT NAME	**An optimal dimensionality reduction technique for cosine similarity**	CLIENT	Ninh Pham ninh.pham@auckland.ac.nz
PROJECT DELIVERABLE	1) An object-oriented variant with Python wrapper (minimum requirement) 2) Test the correctness and efficiency of the released package on standard benchmark data sets (minimum requirement) 3) Make the Python package widely available on PyPI with clear API instructions. 4) Submit pull requests with tweaks to compare CEOs with other competitors on https://github.com/erikbern/ann-benchmarks/		
PREFERRED SKILLS	Experience with high-performance programming in C/C++ (with Eigen, Boost libraries) Experience with GitHub Experience with Docker, Python Good knowledge of randomized algorithms and statistics		

BACKGROUND

This project studies the state-of-the-art solution for nearest neighbor search, an algorithmic component on many computer science applications.

PROJECT DESCRIPTION

The project will implement CEOs, an optimal dimensionality reduction technique for cosine similarity.

Currently, there is a non object-oriented C++ implementation in GitHub. We will implement an object-oriented version with Python wrapper, and test this implementation against state-of-the-art solution (Facebook Faiss, Google ScaNN) on standard benchmark data sets (https://github.com/erikbern/ann-benchmarks/)

AVAILABLE RESOURCES

https://github.com/NinhPham/MIPS
https://github.com/erikbern/ann-benchmarks/
Help from other fellow Nigel Bell

PROJECT 7

PROJECT NAME	**Coderunner Card Sorting**	CLIENT	James Finnie-Ansley james.finnie-ansley@auckland.ac.nz
PROJECT DELIVERABLE	A Coderunner Question Type in Moodle that allows for automatically graded card sorting exercises.		
PREFERRED SKILLS	Web development – HTML, JS, etc.		

BACKGROUND

When learning to program, it is not always useful to JUST answer questions on programming. Different types of questions hone in on different skills. These _could_ be cool question types like being asked to write a program or slightly more boring question types like MCQs. But, there are currently many saucy and exciting question types out there in the world just waiting to be asked – like EiPE, reverse tracing, refute, and parsons problems.

However, due to limitations in software used for lab exercises, it has been difficult to ask these question types in large-scale instruction. That is until a recent update in coderunner that led to an absolute *question bonanza* – providing support for all kinds of juicy and exciting problem types; however, I believe we can push it even further.

Card sorting is an exercise where you organise cards like you do on a Trello board. By carefully choosing cards and the instructions you give people to sort them, you can create interesting exercises both for research and education.

PROJECT DESCRIPTION

The project would involve making a new question type in coderunner (e.g. coderunner.auckland.ac.nz) that allows for automatically assessed card sorting exercises.
This would be a trello-like question type that allows students to organise a set of pre-defined cards into groups.
I already have source code for a few similar projects to serve as a starting point.
This project would support research into card sorting as a new question type and provide a framework for other card sorting studies

The application must work on Moodle using the Coderunner question type plugin.

PROJECT 8

PROJECT NAME	**Sustainable Computing Hub.**	CLIENT	Gerald Weber, Katharina g.weber@auckland.ac.nz, katharina.dost@auckland.ac.nz
PROJECT DELIVERABLE	Implementation in HTML and Vanilla JavaScript and the LAMP stack subject to further specification by project owner. A focus is on good software architecture with as much modularization as possible. Reuse of existing software packages after discussion with the project client is possible and welcome. Thorough testing of the platform is mandatory.		
PREFERRED SKILLS	Knowledge or willingness to learn HTML and Vanilla JavaScript and the LAMP stack.		

BACKGROUND

The energy consumption that is needed for computing processes is an increasing concern.
For example, the energy required to mine Bitcoin could power the country of Finland.
To provide services such as Google search, YouTube, or Gmail, data centres consume enough electricity to power 200,000 homes. Artificial Intelligence has become increasingly power-hungry over the last years making ChatGPT, today's trending chatbot, an expensive application.
As a consequence, sustainability concerns in computing have become a recognized field of study in the community.

PROJECT DESCRIPTION

Here, at our school, we aim to develop an international research hub that allows researchers to share their ideas and solutions on sustainability topics and fosters collaboration.

This project is in collaboration with the Sustainability Flagship Research Project in our school and by the Sustainability Committee of the Faculty of Science.

CONSTRAINTS

Must be written in HTML and Vanilla JavaScript and the LAMP stack subject to further specification by project owner.
The software implementation will be owned by the Sustainability Flagship Research Project

PROJECT 9

PROJECT NAME	**A human-readable question format and authoring system for Moodle/Coderunner questions**	CLIENT	James Finnie-Ansley james.finnie-ansley@auckland.ac.nz
PROJECT DELIVERABLE	A Python program that can transform a suite of Moodle Question data in a format specified by the client to Moodle XML and back.		
PREFERRED SKILLS	Familiarity with Python		

BACKGROUND

The UoA CS department uses Moodle with a Coderunner plugin for labs and assignments in introductory programming courses like CS101 and CS130. However, creating and managing Coderunner questions on Moodle is cumbersome. Having questions managed directly on Moodle also means it is difficult to manage edits to questions and log issues etc. These issues mean managing the quality of Coderunner questions is difficult, particularly in large content-heavy courses.

Moodle does allow for questions to be exported and imported in `Moodle XML` format; however, this format is not intended to be written or edited by people.

PROJECT DESCRIPTION

This project aims to create a system to manage question data externally in a human readable format that can be versioned and managed on tools such as GitHub. The system needs to be able to compile the question data into valid Moodle XML that can be used to import questions into Moodle.

AVAILABLE RESOURCES

It is likely students will need to set up a Moodle instance and Jobe server. The client can assist with setting up local Moodle/Jobe instances for students to test their code or play around with or can provide a Moodle instance.

CONSTRAINTS

Must be written in Python.

PROJECT 10 E

PROJECT NAME	**Terramagotchi - Trōfik edition**	CLIENT	Simon Ingram s.ingram@auckland.ac.nz, shane@actionecology.com
PROJECT DELIVERABLE	A working prototype that builds on the rule-based 'nearest neighbour' approach of falling-sand games to build an abstract, dynamic 2D world containing active elements e.g. plants, bacteria, water, sunlight and carbon.		
PREFERRED SKILLS	Programming, data analysis, capacity to problem-solve and be inventive. Related work supervised/commissioned by one of the clients has involved Java and .js. The client is interested in the work being developed in Unity and ported to web-assembly (as this was a recommendation from last semester's team), or in .js, C# or other language well supported in the UoA eco-system appropriate to the application but which doesn't represent 'over-reach' of the team's technical capacities and allows them to pay proper attention to the translation of the hybrid knowledge of game-based ecological thinking the supervisors are here encouraging and supporting.		

BACKGROUND

Terramagotchi - Trōfik edition draws inspiration from the hidden complex of below-ground exchanges and interactions in soil and the handheld virtual-pet Tamagotchi, released in Japan in 1996.

It is a minimally interactive game-like experience of actors (or elements) in a virtual (2D or 2.5D) plant-microbial system which demonstrates exchanges in soil and captures dynamic trophic (nutrition-based) interactions as well as the effects of sun, water, wind and temperature.

The objective is for the user (or users) to iterate a system by changing its parameters to explore subsequent effects on plant-microbial life in soil.

PROJECT DESCRIPTION

The output is a 2D (or 2.5D) rule-based system employing customised cellular automata-like rules which the team will develop to model the range of different actors in the system undertaking various exchanges and interactions. For instance, the exchange of plant sugars for minerals, predator prey interactions in the rhizosphere and the effects of this on soil structure.

Modifying the application's parameters via a remote application, the user will introduce changes in weightings of actors to affect exchanges and interactions between organisms and their conditions in ways relatively true to reality, albeit in a coarse, or generalised way to offer a iterative, playful, educative experience.

Actors in the system: roots, fungi, bacteria, protozoa, nematodes, micro-arthropods, water, nitrogen, phosphorous, oxygen, carbon, sugar, other minerals.

User actions: tilling, fertilizing, compacting & cutting plants, planting cover crops, adding microbial foods, other actions.

In each instance of the application, a small set of agreed parameters act as initial conditions, including average sun, water, wind and temperature as well as disturbance frequency (tillage/grazing). These parameters are designated by a set of average values (published online/supplied by client) and act as baselines able to be edited by a user through a secondary 'remote' application to affect the system.

The remote is accessed by a smartphone via a QR code in Terramagotchi - Trōfik edition and allows for a minimal set of up to 10 interactions i.e., 'reduce/add synthetic fertiliser,' 'reduce/add rain,' 'increase harvest/grazing frequency,' etc.

This QR code is unique to each instance of the application and is able to be accessed by one or more users. In an art museum this would have the effect of multiple users contributing to the outcome of the system in a collective way.

Expert advice will be given on aesthetic considerations and the nature of the plant-microbial interactions in soil.

Terramagotchi relates to the genre of falling sand games, except in Terramagotchi - Trōfik edition, actors enter the stage from all sides and can also spawn on-stage. In its application of nearest neighbour rules and initial conditions, it also relates to zero-player dimension of John Conway's Game of Life and Christoper Langton's Langton's Ant, except in Terramagotchi - Trōfik edition, a minimal set of parameters can be changed throughout the application's runtime.

Source for the first version of Terramagotchi (developed in semester 2 by Team Teapot, 2023) exists in which a river flows in a landscape featuring trees, rain, worms and rain. Aspects of the design of this application including a version of the remote app described above, will be shared via GitHub.

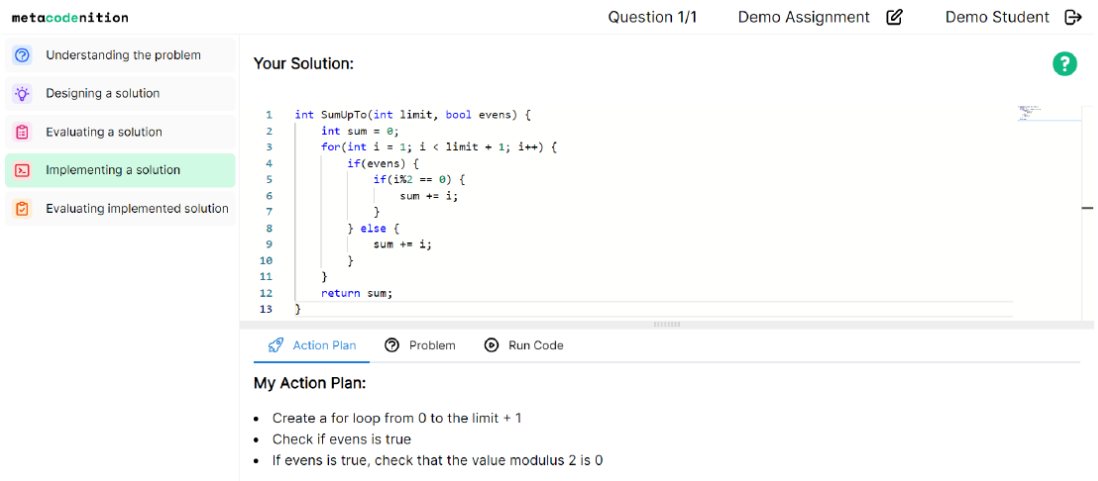
AVAILABLE RESOURCES
Fortnightly supervision meetings by an expert agroecologist (Shane Ward, Action Ecology) and contemporary artist (Dr Simon Ingram, Elam)
CONSTRAINTS
The application must be able to run on a machine without a fast GPU. IP remains the property of Simon Ingram and Shane Ward
RELATED WORKS
• Terrestrial Assemblages (Computer Science students Team Teapot, Simon Ingram), Terramagotchi 1.0: https://terramagotchi.web.app/ • Terrestrial Assemblages (Rikka Ly, Simon Ingram) Falling soil: https://terrestrialassemblages.github.io/Falling-Soil/ • Terrestrial Assemblages (John-Paul Pochin, Simon Ingram), Soil Model, 2020/21 - 8 hours of runtime compressed to 4 mins: https://youtu.be/UORNYiHQbEo

PROJECT 11

PROJECT NAME	**Exploring Novice Interactions with AI Code Generation.**	CLIENT	Paul Denny paul@cs.auckland.ac.nz
PROJECT DELIVERABLE	The primary goal of this project is to instrument a novice-friendly IDE with AI generated code suggestions and explanations, suitable for a beginner. There are many ways that this could be achieved, and an important part of the project will be to consider various options for designing the UI, allowing novices to request code suggestions and explanations, and displaying those suggestions. The prototype tool that is developed should collect fine-grained data on user interactions to support research studies exploring novice interactions with AI code generation models.		
PREFERRED SKILLS	Web development (with a focus on front-end) Interest in UI/UX design Experience working with APIs Interest in large language models (LLMs)		

BACKGROUND

Prior computing education research projects have led to the development of programming environments suitable for novices. The interface of these tools usually includes a code editor, the display of a programming question from a bank of questions, and the display of automated unit-test feedback. One recent example is shown below:



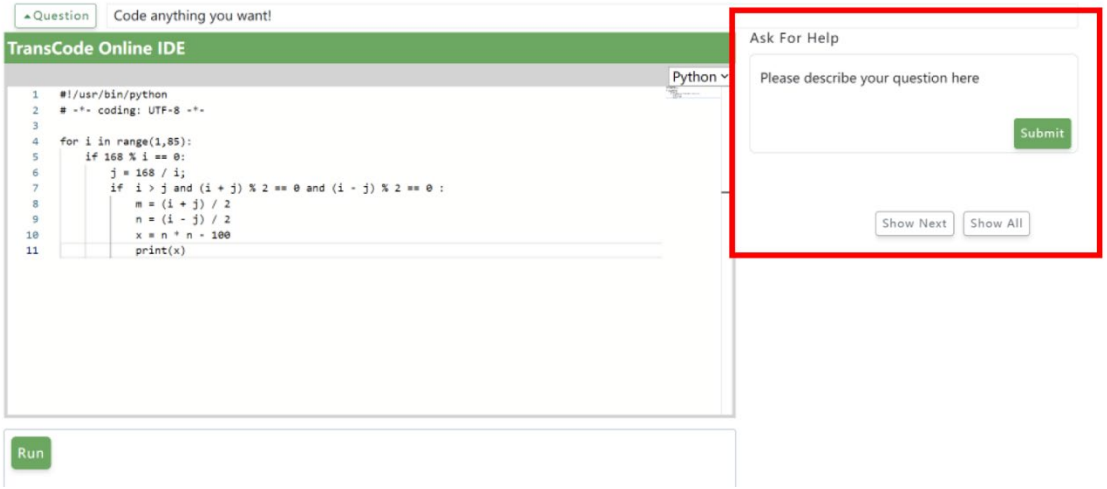
PROJECT DESCRIPTION

This project will involve instrumenting such a tool with the option for students to produce AI-generated code and explanations of code using OpenAI's Codex model (or a similar code-generating large language model).

One popular interface to Codex is the GitHub Copilot plugin, however this plugin generates code suggestions in a way that may not be suitable for novices. Specifically, it attempts to generate code suggestions after each key press, and it often produces suggestions consisting of large blocks of code. For novices, such feedback can be overwhelming and may not be produced in a way that supports learning.

The current project will aim to explore a different interaction model in which students are in control of when to request suggestions from Codex. The project can begin with an existing novice development environment (i.e. from a previous project), or developing this can be included in the scope of the current project.

An example of one (very simple) possible mock-up is below, although a major part of this project will be brainstorming and implementing more interesting approaches.



PROJECT 12

PROJECT NAME	**A web-based tool for organizing markers.**	CLIENT	Burkhard Wuensche (burkhard@cs.auckland.ac.nz) Sudeep Stephen Sudeep.stephen@auckland.ac.nz
PROJECT DELIVERABLE	A web-based tool for organizing markers, which is able to receive student applications, course supervisor requirements, and automatically assigns markers to courses.		
PREFERRED SKILLS	Good programming skills (esp. web-programming, data bases, user interfaces)		

BACKGROUND

Markers are an essential component of many courses. We would like to have a web-based tool for simplifying the marker application and selection process.

PROJECT DESCRIPTION

Course supervisors should be able to specify course relevant information, e.g., expected workload during the semester.

Markers should be able to apply and submit relevant information, e.g., courses they would like to mark (in order of preference), grades, previous experience, current enrolment status, maximum workload, academic transcript and CV.

Course supervisors should be able to select their preferred markers (order of preference).

The tool should automatically assign markers to courses such that coverage of courses is maximized subject to certain requirements (e.g., grades).

The marker coordinator should have access to all components of the tool and be able to overwrite any information and automatic assignment.

PROJECT 13

PROJECT NAME	**A system for programmatically generating animated walkthroughs of data structure operations and algorithms.**	CLIENT	James Finnie Ansley james.finnie-ansley@auckland.ac.nz
PROJECT DELIVERABLE	A tool that can, minimally, support animating generic list operations (e.g. indexing, swapping, updating, etc) that can be composed into algorithms such as Insertion Sort, Hashing/Collision handling etc.		
PREFERRED SKILLS	Familiarity with Python.		

BACKGROUND

Manim (https://www.manim.community/) is a powerful tool that is excellent at creating technical animations and worked examples.

Currently, creating walkthroughs of algorithms like selection sort, priority queue insertion, etc is time consuming.

PROJECT DESCRIPTION

The goal of this project is to develop a system that can programmatically generate animations for these algorithms with explanations for each step.

This will hopefully drastically improve the programming examples we have at UoA and will create an amazing resource that instructors can use to teach data structures and algorithms.

PROJECT 14

PROJECT NAME	**A web-based tool for providing marking support**.	CLIENT	Ewan Tempero e.tempero@auckland.ac.nz
PROJECT DELIVERABLE	A web application that satisfies the requirements outlined in the project's description.		
PREFERRED SKILLS	REST and Javascript		

BACKGROUND

Many people use spreadsheet to support marking assignments. Spreadsheets are error-prone (e.g. easy to accidentally type into wrong cell, hard to navigate submissions with respect to rubric) and tedious to set up (even if the rubric is the same as last year, can't just use last year's spreadsheet because various details have changed, such as Canvas identifiers).

Many assignments have marking rubrics, but they are presented in different ways (e.g. basic text files, html, email instructions) and cannot be easily linked to the spreadsheet with the marks (e.g. part way through the marking, if you realize that some part of the rubric needs to change, it is not easy to find all the relevant assignments where the change applies).

PROJECT DESCRIPTION

This project would be more than just a "rubric creation tool' (of where there are a number available.
It would be more than providing a means to enter marks (like speedgrader), or to provide feedback comments/marking (like Inspira). The goal is to do something like all of these but do it right. Specifics:
* support upload of Canvas spreadsheet
* support download to Canvas spreadsheet
* support rubric development and modification
* support navigation of submissions by rubric

A random search for what's available yielded this:
https://www.profweb.ca/en/publications/articles/grading-made-easy-digital-tools-to-create-rubrics
which looks representative (but not necessarily definitive) of what could be done.

PROJECT 15 E

PROJECT NAME	**An online portfolio to showcase capstone projects.**	CLIENT	Asma Shakil asma.shakil@auckland.ac.nz
PROJECT DELIVERABLE	An online web application that showcases capstone projects.		
PREFERRED SKILLS	Full stack web development.		

BACKGROUND

The computer science capstone course provides a golden opportunity for students to showcase their skills to potential employers. Students work in teams of 5 – 6 people on a substantial project that aims to solve a real-world problem. Currently, we hold an in-person showcase event at the end of each semester, where we invite industry representatives, staff, friends, and family members to see the capstone project outcomes. However, this approach is limited in its outreach.
Our goal in this project, is to develop an online website for the capstone course which showcases students' projects to a wide audience, including prospective employers, clients, and other students.

PROJECT DESCRIPTION

The MVP for this project would be a website that has the following functionality:

1. A gallery showcasing student projects presented in a visually appealing manner, (e.g., https://devpost.com/software)
2. An ability for students to login and specify their project's details (e.g., https://devpost.com/software/surg-e-lite)
3. An ability for visitors to login in, leave comments, and like projects.
4. An ability for the teaching team to login and award badges to the winning projects.

Beyond the MVP, the website could be expanded to include other functionalities needed for the course such as

1. Letting students upload their profiles including, skills and interests that can help them find teammates.
2. An ability for students to discover teammates to work with (perhaps some kind of filter to search by passion/interests),
3. A project bidding system that allows students to specify their top *n* project choices.
4. Any other functionality that you see important as a student enrolled in this course :)

PROJECT 16

PROJECT NAME	**Water Cycle Model**	CLIENT	Simon Ingram s.ingram@auckland.ac.nz
PROJECT DELIVERABLE	MVP: 3D dynamic model of the water cycle in the Taranaki region. Default POV fly-through. The system designed with an eye that river water quality data can be integrated into the system by a non-specialist administrator via a webform in a later phase. Phase 2/MVP extension: Viewer-user able to interact with the model using a limited number of navigational options. An administrator able to enter river quality data from key sites and for this be integrated into the model and expressed in the colour values of given rivers.		
PREFERRED SKILLS	Programming, data analysis, capacity to problem-solve and be inventive. The client is open to different programming environments supported by the UoA eco-system appropriate and to the application. The client is interested in the work being developed in Unity as he understands this might provide the capacity for the application to be exported to different formats for different devices. A key concern is that the choice of programming environment doesn't represent 'over-reach' of the team's technical capacities and allows them to pay proper attention to the translation of basic earth science principles programmatically.		

BACKGROUND

This is an art-science project demonstrating a generalized model of the circulation of water in Taranaki's earth-atmosphere system.
It utilises programming and data analysis methods to tell the story of the processes of evaporation, transpiration, condensation, precipitation, and runoff in the Taranaki Rohe.

PROJECT DESCRIPTION

A viewer-user will experience a full screen render of a 3D model of Taranaki modeling the dynamic processes of water. The model will be built-up by a combination of procedural generation methods and publicly available data derived from in-situ, remote sensing and Earth observation approaches.

As the project develops, a sense of what data is publicly available, and what can be inferred from this data, will help define 'realism' in the model - the extent to which water processes are generalised/made particular to actual data. By default, the user-viewer's point of view (POV) will be a fly-through over the land as if from the perspective of a great bird able to fly great distances at different altitudes.

A remote app is accessed by a smartphone via a QR code presented in the lower left corner of the application. The app allows for a minimal set of navigational options related to POV that would be taken up by one user-viewer in a gallery setting at a time via a smartphone.

In an earlier project (sem 2, 2022) an implementation of this kind of remote app was developed in the Terramagotchi Project by Team Teapot computer science students using Firebase for a .js web app (see Terrestrial Assemblage's Github repo in 'related works/resources' below). The application's focus is a demonstration of the water cycle but the client would like to see groundwork laid for the anthropogenic effects of agriculture to be visualised including: nitrate contamination & turbidity in rivers (where data is publicly available), two-pond systems (visible in satellite images), discharge permit locations (publicly available).

MVP: 3D dynamic model of the water cycle in the Taranaki region. Default POV fly-through.
The system designed with an eye that river water quality data can be integrated into the system by a non-specialist administrator via a webform in a later phase.

Phase 2/MVP extension: Viewer-user able to interact with the model using a limited number of navigational options. An administrator able to enter river quality data from key sites and for this be integrated into the model and expressed in the color values of given rivers.

AVAILABLE RESOURCES

Fortnightly supervision meetings contemporary artist Dr Simon Ingram, Elam

CONSTRAINTS

IP remains with Terrestrial Assemblages, code is made open-source.

RELATED WORKS

Water cycle: https://www.britannica.com/science/water-cycle
Terrestrial Assemblages: https://github.com/terrestrialassemblages
Terrestrial Assemblages, Soil Model, 2020/21 - 8 hours of runtime compressed to 4 mins: https://youtu.be/UORNYiHQbEo
Earth observation technology in New Zealand: https://tinyurl.com/35vc2ty7
https://data.linz.govt.nz/layer/107438-taranaki-lidar-index-tiles-2021/