ISSP Comp 3800 Burnaby Excel Catalogue (2026 Winter)

Project ID:
977

Company:
BCIT MRI Program

Project Areas:
empty

Company Profile:
The aim of the Magnetic Resonance Imaging (MRI) program is to prepare graduates with the knowledge, critical thinking and technical skills to safely perform high quality diagnostic MRI studies while maintaining a high level of care for the patient.

Project Description:
Ray collaborated with Computing students—alongside Vienna (coder/multimedia developer) and Jason (3D modeler)—to advance the development of a VR platform simulating an MRI department. To date, the team has built the virtual environment, including key components such as MRI coils, a power injector, and a patient avatar with limited movement capabilities. A new proposal is being submitted to support continued development, as this project was originally envisioned as a multi-term initiative. The next phase will focus on addressing a bug identified during recent QA testing, with key priorities including correcting pivot points, adjusting scale, and optimizing textures.

Programming Language(s):
Unknown at this time what students were using

Hardware/Software Requirements:
Headsets (supplied by SOHS Sim team) Software-students have been using Unity

Current Work/Arrangement:
Fall term 2024- Ray and Computing students worked together (along with Vienna, coder/multimedia developer; Jason, 3d modeler) to develop a VR platform to simulate an MRI department. Environment has been built so far (coils, 3D models for MRI/power injector), along with patient avatar (some movements).

Previous Project?:
yes

Project ID:
1023

Company:
Gigsup AI

Project Areas:
empty

Company Profile:
Gigsup is an innovative platform designed to help youth navigate career paths through transparent insights on jobs including pathways, education, ROI, demand, skills, and other real-world data (including, pay, stress, demographics and more). Our platform will empower users to make informed career choices by delivering data-driven insights with minimal cost. We're seeking a tech-savvy BCIT student with a passion for developing impactful solutions to help us build our MVP!

Project Description:
Gigsup is a tech-driven platform that helps youth explore career paths through transparent, real-world data; everything from job demand, required skills, and pay, to ROI on education, stress levels, and workforce demographics. We're currently enhancing our MVP and looking for a BCIT student who's passionate about using technology to make an impact. Our MVP matches users (based on resumes, LinkedIn, and survey data) to job profiles aligned with their unique strengths. We now want to level up: • Build an open profile network with search, mentor/mentee assignment, and interaction capabilities • Scrape job postings and datasets to match users with real, live opportunities • Strengthen our backend to support a scalable LLM that connects skills to career pathways, and then to education and job opportunities If you're excited to build meaningful tech that solves a real problem, we'd love to hear from you.

Programming Language(s):
Python / Llama

Hardware/Software Requirements:
Cloud Platforms TBD, Data Storage TBD, NLP Library TBD, Data & Analysis TBD

Current Work/Arrangement:
Career Assessment: The initial assessment model has been developed to provide career recommendations. Plans are underway to refine this model by incorporating psychometric and skill-based evaluation metrics to enhance personalization. User Profiles: Structured user profiles are being designed to capture individual skills, experiences, preferences, and career aspirations. The goal is to implement AI-driven insights that suggest potential career transitions and skill development pathways. Career Pathway Recommendations: The LLM is being trained to map CV content to relevant career paths and educational opportunities. Efforts are focused on providing dynamic recommendations based on real-time job market trends and integrating alternative career paths, including entrepreneurship and freelance opportunities. Job & Education Links Integration: Work is in progress to connect career pathways with real-time job postings from industry partners and job boards. Additionally, the team is linking recommended pathways to relevant educational courses, certifications, and skill-building programs, providing direct application options and enrollment links for job and education opportunities. Data Architecture: The backend database is being optimized for scalability and seamless integration of new data sources. Measures are being implemented to ensure GDPR and privacy compliance in handling user data, along with feedback loops for continuous learning and refinement of recommendations. Next Steps: Conduct user testing on the initial LLM prototype to assess its effectiveness. Gather industry feedback to validate and refine career pathway recommendations. Develop a Minimum Viable Product (MVP) showcasing the AI-driven career guidance experience. Prepare for integration with external job and education databases.

Previous Project?:
yes

Project ID:
1058

Company:
iron52

Project Areas:
Web Development,Mobile Development

Company Profile:
We make mobile apps to help people improve their lives. Two people: one IT professional and one legal professional.

Project Description:
We are making a mobile app whose interface is like that of Tinder. It is not a dating app, but the experience is very similar to the look, feel, and behavior of tinder. The app will allow people to find "things to do" in their area, in their budget, on the days they are available.

Programming Language(s):
Firebase Swift and SwiftUI with XCode

Hardware/Software Requirements:
A Mac capable of running xcode.

Current Work/Arrangement:
We are just starting.

Previous Project?:
no

Project ID:
1077

Company:
Cherry FLFM Inc.

Project Areas:
Mobile Development

Company Profile:
Cherry FLFM Inc. is a Vancouver-based startup founded by and for sapphics—queer women and non-binary people. Our mobile app, Cherry FLFM (Find Love, Friendship, and More), is designed to provide a safe, inclusive space for sapphics to build meaningful connections, whether romantic or platonic. While mainstream dating apps exist, they often fail to serve this community, exposing users to harassment, discrimination, or irrelevant matches. Cherry FLFM addresses these gaps through advanced filtering, robust safety features, and thoughtfully designed profiles that center queer users.

Project Description:
Students who select Cherry FLFM will gain hands-on experience maintaining and developing a live mobile app with active users. Cherry FLFM is a friendship and dating app built specifically for queer women and non-binary people, with a focus on safety, inclusion, and community. This project will involve maintaining and upgrading Version 1 of the app. Key tasks may include resolving bugs, improving performance, and optimizing the user experience. Students will also contribute to planning and developing new features for Version 2, including the introduction of paid memberships and premium community-focused features. Students will have the opportunity to: - Work on a real product that is actively launching - Apply and expand their Flutter, Firebase, and mobile development skills - Collaborate with a queer-led team deeply invested in ethical tech - Make a tangible impact in a community often underserved by dating and friendship apps The sponsor is actively involved, has technical support available, and is interested in building a long-term working relationship if the collaboration is successful. This is a meaningful opportunity to apply your skills to a real-world app that fosters connection and belonging in the queer community.

Programming Language(s):
The app was built using Flutter, but I am open to using any programming language as long as it is compatible with iOS and Android.

Hardware/Software Requirements:
Any hardware/software that is required to code a mobile app.

Current Work/Arrangement:
We hired a development team in India to build Version 1 of the Cherry FLFM. Cherry FLFM is available on the Apple and Google Play stores.

Previous Project?:
no

Project ID:
1085

Company:
Soul Surfer Software by MTI

Project Areas:
empty

Company Profile:
We are a software startup specializing in small projects, community software and digital support work for BIA's (business improvement areas). Are history is eclectic as we started as a community restaurant 25 years ago in North Vancouver. Our community involvement led us to see technology needs at the community level. We have done a project with BCIT called TipShare. It is a tip distribution calculator and fintech solution. Currently 3 BCIT grads are working on the application, and we hope to go live very soon.

Project Description:
This project is a community mobile application and PWA (time permitting). We are working closely with the Central Lonsdale BIA (business improvement are) to develop a mobile first digital solution for their upcoming charter to represent and develop local businesses. The MVP is a business listing for each business in the hyperlocal area of their charter and up to 4 event listings that individual businesses could enter into the app through forms. These listings need basic functionality and filtering as well as way finding. As the app is partially developed already, the task would involve improving and completing the application and going through the App Store and Google Play protocols to first launch the beta version and second go live with the application. There also needs to be a PWA (progressive web app) configured so the task (time permitting) would be to assess and configure the UI/UX for different screen sizes. This MVP goal is a launchpad for more sophisticated modules such as hyperlocal delivery, hospitality reservations, peer to pier payments and more. This app is the first of its kind and aims to be eventually launched across Canada for all BIA's. There are currently more than 800 BIA's in Canada. This MVP is beta version with Central Lonsdale as our Beta group. You will work in React Native and Xano database. The social goals of the app are two fold: 1. It should provide small business technological tools to better compete with the large corporations that are dominating the business landscape and 2. Encourage local residents to choose and support their local merchants and businesses by providing those merchants with a technological advantage that they currently do not have.

Programming Language(s):
Draftbit is a no code and/or custom code platform. We suggest using the no code structure with custom coding when it suits your needs. Xano is a no code database. We have notes from previous teams to help with 'quick start'.

Hardware/Software Requirements:
DraftBit and Xano will be provided

Current Work/Arrangement:
This is described above. There is a current working copy of the app. It is basic but has both the database and frontend substantially completed.

Previous Project?:
no

Project ID:
1096

Company:
Accountium INC

Project Areas:
DevOps,Web Development,Other

Company Profile:
Building Accounting Software and the AI Accountant

Project Description:
Accountium is building Accounting Software and AI technology related to Accounting. We are looking for UI to improve the look of the Cloud Application and mobile version optimization; web designer for our presentation website, programmers for various modules inside the application.

Programming Language(s):
C#, PHP, HTML, Java

Hardware/Software Requirements:
laptop, PC, printer

Current Work/Arrangement:
The Team leader puts a request for task in the Azure devops and the developer get's the task from there, push the code in the main branch and mark the task complete.

Previous Project?:
no

Project ID:
1113

Company:
Soul Surfer Software

Project Areas:
DevOps

Company Profile:
We are currently developing three apps: financial analysis, community based software and tip sharing application.

Project Description:
This project is a continuation of a project that began as part of a ISSP project at BCIT in January 2025. The app's name is Equitip, and it is a tip sharing calculator and distribution app along with a PWA (progressive web app). It uses adjustable settings and mathematics to allow staff tip pools to customize their tip distributions to create fair distributions among members. EquiTip is currently published at the App Store and Google Play Store. The next phase of the project is to complete the PWA version of the app and complete the development work to install the fintech back end for the app. We will be using a Canadian company as our fintech back end to provide distributions from businesses to their staff members directly.

Programming Language(s):
React Native is primary language

Hardware/Software Requirements:
We are currently using Draftbit (no code/custom) for the front end and Xano for the backend. While a Mac workstation is simpler to set up, PC work stations are possible for the builds.

Current Work/Arrangement:
Tasks are currently carried out remotely. We use discord for communication and meeting and work tickets are distributed by the team lead

Previous Project?:
yes

Project ID:
1133

Company:
IATSE Local 891

Project Areas:
Web Development,Mobile Development

Company Profile:
IATSE 891 consists of over 10,000 professional artists and technicians who work in film and television production in BC. We support all genres of production at all budget levels and have made some of the biggest feature films and some of the longest running and most successful television series ever screened. From our earliest days, we have actively participated in the development and growth of the film and television production industry in BC and marketed our province as a premier filming destination. Our approach is to supply world class talent and act reasonably to ensure a stable labour relations climate within our industry. IATSE 891 provides its members with an extended health and dental plan, an RRSP plan, an employee and family assistance plan, and access to other benefits. We also provide our members with ongoing training with a strong emphasis on safety.

Project Description:
IATSE 891 is committed to advocating for whole person safety for every worker. Through this project, we want to establish a stand alone web portal for 891 members to be able to record and report incidents of unsafe work, harassment, bullying, or workplace violence. We have branding and content prepared, and a general structure in mind, so this project would focus on creating the technical platform to allow members to submit full reports for investigation(with personal contact information), submit a report anonymously, or reach out for help to a steward or ombudsperson. We have several reference examples from other similar projects run by other unions in the US, and a clear list of project milestones and deliverables. The leads internally are Hillary Bergshoeff (Assistant Business Representative) and Quentin Bennetti (Communications Coordinator). Our ideal working partner(s) would be a team committed to growing worker safety and power at work, and empowering rank and file workers with the ability to hold their Employers accountable for providing a safe workplace for every worker.

Programming Language(s):
TBD

Hardware/Software Requirements:
The portal should be accessible for any member, via any mobile, tablet or desktop device. The portal should also be designed for maximum compatibility with screen readers, and other accessibility supports - particularly important for this project as our research has made it clear that workers with intersectional underrepresented characteristics and disabilities are at significantly higher risk of workplace harm.

Current Work/Arrangement:
If we are not chosen here we will likely have to wait to develop the web portal until the 2027 budget year and send out an RFP to market.

Previous Project?:
no

Project ID:
1135

Company:
KOM Community Policing Centre

Project Areas:
Mobile Development

Company Profile:
KOM Community Policing Centre (KOM CPC) is a volunteer-driven organization committed to enhancing safety and connection in Vancouver's neighbourhoods. We work closely with a range of stakeholders, including residents, local businesses, schools, and social agencies, to deliver effective crime prevention, public education, and community engagement programs. In partnership with the Vancouver Police Department (VPD), KOM CPC helps bridge the gap between law enforcement and the community, fostering collaboration, trust, and proactive solutions that promote lasting neighbourhood well-being. Our inclusive approach ensures everyone has a voice in creating a safer, more vibrant community.

Project Description:
ChildGuard ID Mobile App Project Description ChildGuard ID is an innovative mobile application developed for the KOM Community Policing Centre (KOM CPC) to enhance child safety within the community. The app allows parents and guardians to securely store and manage essential identification information for children directly on their smartphones. The primary development focus is on the iOS platform, with potential Android support if time and resources permit. Project Goals - Develop a secure iOS mobile application that enables families to save a child's physical description, medical details, and photographs using local device storage. - Facilitate rapid emergency communication by enabling quick sharing of child profiles with trusted contacts and community policing centre personnel. - Incorporate location feature so parents can attach location information to profiles and share updates efficiently. - Ensure strong privacy and security by prioritizing local storage of sensitive data with optional future cloud integration. - Design a user-friendly interface tailored for easy and quick access in emergency situations. Community Context This project supports the mission of KOM CPC to keep children safe in the community. While KOM CPC works closely with the Vancouver Police Department (VPD), this app is an independent initiative of the community policing centre. Once completed, ChildGuard ID will be available to any Community Policing Agency, enabling broader use and impact in improving child safety across communities. Expectations for Student Teams - Technology choice freedom: Students may use any iOS development frameworks or tools they prefer. - Implement secure and reliable local data storage with documentation on potential cloud tie-in options. - Develop profile management and secure sharing capabilities suited to real-life emergency needs. - Conduct usability testing with input from KOM CPC representatives. - Provide detailed technical and user documentation. - Engage in regular communication with KOM CPC project sponsors for feedback and guidance. Value & Impact ChildGuard ID will be a valuable tool for families and community policing centres, enabling faster and safer responses when children go missing or require immediate assistance. This project offers students hands-on experience in mobile development, data privacy, and creating solutions with tangible community benefits.

Programming Language(s):
Freedom in tool selection: Students may use any iOS development framework or toolkit (e.g., SwiftUI, UIKit, React Native, Flutter for iOS) based on their strengths and interests.

Hardware/Software Requirements:
App just needs to run on Apple iPads/Phones. We do have an Apple Developer Account.

Current Work/Arrangement:
The KOM Community Policing Centre (KOM CPC) manages child identification information through a paper-based process. Historically, much of this work was supported by Child Find BC, an organization dedicated to helping locate missing children. Child Find BC provided forms, collected information, and offered guidance on swift emergency response. However, Child Find BC has since gone out of business, leaving a gap in local child safety resources that now needs to be addressed. Currently, these tasks are carried out as follows: - Collecting Information: Parents and guardians fill out paper forms provided by the CPC, noting the child's physical description, medical details, and recent photographs. These forms were originally distributed and managed with Child Find BC's guidance. - Storing Records: The completed forms are securely stored on CPC premises in physical filing systems. - Updating Information: Any updates or changes require families to visit the centre in person, resubmitting new details and photographs. - Access and Sharing: In case of an emergency, CPC staff must manually retrieve and copy paper records, then share information with police or search organizations. Previously, Child Find BC would coordinate these responses; that centralized support no longer exists. - Status of Data: The information is only available in physical format, limiting accessibility, timely updates, and rapid sharing with authorities. With Child Find BC no longer operational, there is an immediate need to modernize and digitize the way child identification profiles are managed. A digital solution will ensure information is securely stored, easily updated, and quickly shared with police and community agencies, filling the void left by Child Find BC and helping keep children safer in Vancouver and beyond.

Previous Project?:
no

Project ID:
1140

Company:
Being Love Incorporated DBA The GOOD GOODBYE

Project Areas:
Web Development,Machine Learning/Algorithms/Research

Company Profile:
The Good Goodbye is a new company that is operating as a one stop shop for death related services and educational opportunities for the public. Currently we offer quarterly Symposiums bringing on experts to speak about different things like preplanning funeral arrangements, to the role of a death doula to how to process grief and deal with family disputes as they come up. We are looking to get a platform built for the general public so they dont get ripped off at a vulnerable time and can be told of what their choices actually are every step of the way. We are also focused on Cultural conversations and how to balance family disputes on how things should go so that everyone ends up satisfied. There is a huge need for transparency in the industry and we are here to help people through this difficult time. We are looking to create interactive AI components to the website. We are looking to different subscription levels and be affordable for anyone to commit to. (looking at 3-5 dollars a month as a first level base).

Project Description:
We are looking to create a subscription platform where people can interact with AI technologies to get help with funeral planning, a marketing platform for death related service providers, and a place where people can get information on whatever they are dealing with when it comes to death. We are looking to develop an interactive space where community can come together. We are also aiming to create a secure space for people to put memory documents into a secure server that is able to hold video wills, will drafts and any private things they would like to store safely. We really just need to get the foundations set up for everything at this point. I (Kimberlee Klein) have worked with development teams before in previous years for other unrelated projects and have a large bank of code available if the students want to re-use any of the code that will help move this project faster.

Programming Language(s):
You can choose. The database of code available from Camosun college students was React, JS node.

Hardware/Software Requirements:
Not sure.

Current Work/Arrangement:
Currently we have Replit set up and no back end has developed for real (so its deemed useless :)) I would like a student development team to decide what would be best to use in real life for developing this platform. i do have access to a huge amount of code if the students need ideas or can reuse code from other years.

Previous Project?:
no

Project ID:
1147

Company:
Grant Hochman

Project Areas:
empty

Company Profile:
I am submitting as an individual.

Project Description:
Project: To create an app (or software) that will record and transcribe an unknown language. In other words, it will record and transcribe unknown or very unusual words and phrases. Observation and reason for the app or software need: I am a caregiver for my wife with Alzheimer's disease. While she still has a few phrases in English, when she speaks her own “language” the facial expressions and tone of voice suggest this is both a real and meaningful language for her. I would like to have the opportunity to repeat back various portions of her language to see if this brings a knowing response. If so, this will provide one more means of communication. Everyone suffering from this disease will have their own “language.” It's estimated that over 55 million people are suffering with this terrible disease worldwide, so if successful, this could provide some unexpected help to many people. Need: Currently I’ve not been able to locate an app or software for this situation: The present apps only record and transcribe known languages. If it's unknown, the app. will guess at equivalent sounding English words, rendering it useless. I want to add that if it’s too difficult to record and transcribe, then transcribing only is to be preferred. But I note that the recording allows one to hear the tone of voice and emphasis.

Programming Language(s):
No coding is required.

Hardware/Software Requirements:
The creation of an app or software will be needed.

Current Work/Arrangement:
As per the project description above, no app or software currently exists that can record and transcribe an unknown language.

Previous Project?:
no

Project ID:
1148

Company:
Element Health

Project Areas:
Mobile Development,Games,Machine Learning/Algorithms/Research

Company Profile:
At ElementHealth, we are driven by a shared passion for improving well-being through technology and Traditional Chinese Medicine (TCM). Our team brings together expertise across data analytics, business strategy, and product development, allowing us to create solutions that are both evidence-based and user-friendly. With strong backgrounds in research, software development, and project management as well as the support of our partners PBY Ventures, we combine global perspectives and practical experience to deliver a product that bridges the gap between ancient wisdom and modern lifestyles. Collectively, we've worked on initiatives ranging from data science and analytics to healthcare operations and education. This diversity enables us to design tools that not only track personal health but also empower users with actionable insights. What unites us is a commitment to making health accessible, engaging, and meaningful. We believe that health is more than just numbers—it's about daily balance, habits, and human connection. ElementHealth was built on this belief, and our team is dedicated to helping users take small, consistent steps toward a healthier and more harmonious life.

Project Description:
We are looking to add a few gamification features to our existing apps (Android and Apple). The idea is that a user will be encouraged and rewarded for continuing healthy habits which will overtime help them by reinfocing positive behaviors.

Programming Language(s):
PRimarily javascript

Hardware/Software Requirements:
The students will need access to both an android and apple device for eventual testing which we can provide but really there are no significant hardware requirements.

Current Work/Arrangement:
We have two apps in production which we developed with the help of our dev partner. We are looking to add new features.

Previous Project?:
no

Project ID:
1150

Company:
Granville Biomedical Inc.

Project Areas:
Mobile Development,Machine Learning/Algorithms/Research

Company Profile:
Granville Biomedical is a Canadian health-tech company specializing in the design of women’s anatomical pelvic health models to enhance clinical practice and advance patient education. The life-like anatomical training models designed by Granville Biomedical can be used to rehearse routine exam and surgical procedures. Through interdisciplinary research, collaborative design and execution, Granville Biomedical aims to advance international medical education and healthcare training by improving access to advanced simulation learning opportunities and anatomical models.

Project Description:
The Granville Biomedical app provides educational content relating to women’s pelvic health and our anatomical models to our customers. A preliminary version of the app, including video tutorials and product information, was submitted to the App Store and Google Play Store by the Fall 2025 team. We are now looking to develop the next iteration of the app. Priorities for the next iteration of the app include: - Expanding out the AI portion of our app by developing learning modules on various pelvic health topics for iOS and Android - Setting up In-App Payment for subscription users - Submitting the updated version of the app to the App Store and Google Play Store

Programming Language(s):
Previous BCIT students have used: - JavaScript - React Native

Hardware/Software Requirements:
Previous BCIT students have used personal hardware such as laptops and smart phones. At least one team member will need to have iOS devices for testing. Previous students have also used the following: - Github - Visual Studio Code - Android Studios - Expo CLI - Firebase - Xcode - Npm - Trello - Figma

Current Work/Arrangement:
Granville currently does not have a development team. All application work has been conducted previously by BCIT students.

Previous Project?:
yes

Project ID:
1153

Company:
WorkSafeBC

Project Areas:
Mobile Development

Company Profile:
WorkSafeBC is a provincial workers compensation organization that supports workplace health and safety, injury prevention, and return to work. We provide services to employers and workers across British Columbia and develop tools and resources that promote safer worksites. Our Innovation team partners with industry, researchers, and educators to explore new ideas and create solutions that improve safety outcomes.

Project Description:
Fall protection plans are required on many construction and maintenance worksites, but employers often rely on paper forms, printed drawings, or hand drawn sketches. On a worksite, people almost always have a phone, but they may not have access to a printer or laptop. This project invites students to design and build a mobile app that makes it easy for employers and workers to create a fall protection plan directly on their device. The app should guide users through the key sections of the existing WorkSafeBC Fall Protection Plan template. This includes entering basic site information, describing work at heights, identifying hazards, selecting the type of fall protection system, and documenting rescue procedures. The template also includes a diagram page where employers sketch anchor points, roof edges, access points, and hazard locations. The mobile app should support this by allowing users to upload or take a photo of a site drawing, or use a map view, and then place anchors and hazards using simple drag and drop tools. Key features may include: * A guided workflow that follows the Fall Protection Plan template * Entry fields for site details, hazards, selected systems, and procedures * Uploading or photographing a drawing or map * A map or overhead image layer * Drag and drop placement of anchors, access points, and hazards * A digital signature or initial feature so workers can confirm they have reviewed the plan * A simple export option such as PDF or screenshot The domain is workplace safety and injury prevention. The template defines what must be included in a fall protection plan and provides the structure the app should follow. We can supply the template, explain each section in plain language, and help the team understand why certain information is required. We also believe students may have an advantage in understanding how younger workers think and engage with digital tools. We welcome their insights and creative approaches to designing an app that is intuitive, modern, and effective for today’s workforce. The project has an achievable and flexible scope. Students may take on the full end to end workflow or focus on a subset such as the diagram interface, the guided entry form, or the digital signature and export capability. We will collaborate with the team to set milestones based on the skills and interests of the students. Our team includes both technical and domain experts who can answer questions about workplace safety requirements, mobile app design considerations, and general development approaches.

Programming Language(s):
We are open and flexible to using the best technology for the project. Internally, we work with Microsoft .NET and Microsoft Azure, but students can choose the technologies they are most comfortable with. Python is acceptable for any AI related components or data handling.

Hardware/Software Requirements:
No special hardware is required. Standard laptops or desktops are sufficient. We cannot provide access to our Azure environment. Students may use their preferred development tools, libraries, and platforms. We can provide the Fall Protection Plan template and guidance on our general technology environment and domain knowledge.

Current Work/Arrangement:
This is not applicable.

Previous Project?:
no

Project ID:
1154

Company:
Being Love Incorporated DBA Woo Woo Network

Project Areas:
DevOps,Web Development,Scripting/IT

Company Profile:
I have a functional website that needs to be beta tested with users to ensure bugs and protocols are taken care of. Woo Woo Network is a platform where healers are connected to clients. The idea is that healers suck at marketing and collecting money for their services. Woo Woo Network is here to connect clients to healers and also showcase real life results on what the healer has accomplished with each client so as to make us understand that nature of impact the healer is capable of. Mental health issues are addressed easily.

Project Description:
I have been working with students to develop this in the past. It is finally ready to be beta tested for the real world and I am already aware of systems that have not been thought out yet such as when money is received on Stripe, how to set up the healer to ensure they are automatically getting paid out. I know there are more processes that need to be tackled and solved before a major launch of these services. This opportunity would bring students together to launch a web based application and work through bugs as needed and set up a updating software processes that can potentially be automated. Also there may be an opportunity to create new features for the app if there is seen a need.

Programming Language(s):
React, Js Node, I am not a techy. AWS. I do have a tech person available to connect with the students.

Hardware/Software Requirements:
React, JS Node, AWS,

Current Work/Arrangement:
Currently the website is officially up and running. It has not been beta tested yet. It is sitting on AWS.

Previous Project?:
yes

Project ID:
1155

Company:
Being Love Incorporated DBA Leftovers For All App

Project Areas:
DevOps,Web Development

Company Profile:
Leftovers in our home often get thrown out. Everyone feels bad when they throw food away. Im out to get rid of Middle Class Guilt! Before leftovers can turn bad, we can look in our fridge and put a picture of our perfectly good food that you will not eat on time, onto the app. A neighbor can look on the app and see that perfectly good food is available and arrange to pick the food up. When food is picked up, the giver will feel like a hero! We are also out to help families make ends meet cus food is a serious matter. Feeding your family is a serious matter. I have a Leftovers app that has already been developed and needs to be beta tested and put onto the app store properly. We are looking to do this as a project that can be launched into the world by end of April and in the project we will have onboarded at least 60 users.

Project Description:
Looking to launch a Leftovers For All app that makes a huge difference in real life for real users. Going through how to set up the app into the Play Store and App Store is part of the scope. Also dealing with bugs and seeing where improvements can be made in code will be vital to this project's success. We are looking to onboard at least 60 users by end of this project and to receive feedback from users that will inform improvements to be done by end of this semester (April 2026). The ultimate goal is to have this ready to go and ready for launching fully into communities by May 2026.

Programming Language(s):
flutter based, dart, java-kotlin, javascript/ node.js

Hardware/Software Requirements:
flutter, android studio, git, IOS firebase, gradle dart sdk

Current Work/Arrangement:
Currently i have code done and need the app to be launched properly in the app store. there are packages that need to be updated as the code is from 2023.

Previous Project?:
yes

Project ID:
1157

Company:
N/A

Project Areas:
Web Development

Company Profile:
This is a non-profit entrepreneurial initiative led by Yaniv Talmor. The project focuses on democratizing high-level financial planning tools for independent life insurance advisors. We are building a "Financial Needs Analysis" (FNA) web application that helps advisors calculate complex estate liabilities, tax burdens, and insurance gaps for their clients. The project operates in a modern, agile environment, aiming to replace archaic spreadsheet-based workflows with a secure, cloud-native SaaS solution.

Project Description:
Project Title: Financial Needs Analysis v2.0 – AI-Integrated InsurTech SaaS Description: We are seeking a skilled team of Full-Stack Web Development (FSWD) students to architect and build Version 2.0 of a validated financial modeling platform for life insurance advisors. The application automates the complex math required for estate planning and corporate risk analysis, and integrated AI to author documents and improve compliance. Students will work with an existing v1 codebase (built in Next.js/Drizzle) as a reference specification to build a production-grade v2.0. Key Modules to Build/Refactor: Complex Financial Engines: Implement calculators for "Settling Requirements" (Probate, Final Taxes), "Income Replacement," and "Corporate Shareholder Analysis" (EBITDA contribution, Key Person risks). Interactive Data Dashboard: A visual report dashboard using Recharts to display Net Worth, Tax Burdens, and Liquidity scenarios. GenAI "Co-Pilot": Integration of LLMs (OpenAI) to analyze client financial data and auto-write compliance documents ("Reasons Why" letters) and cover letters. SaaS Infrastructure: Implementation of multi-tenancy via Kinde Auth, subscription gating via Stripe, and secure document handling via UploadThing. Competitive Analysis & Market Gap: The Canadian market is currently served by legacy incumbents such as: - Life Design Analysis (LDA) (https://www.lifedesignanalysis.com), - Equisoft/plan (https://advisor.equisoft.com/products/plan), - RazorPlan (https://razorplan.com), - Snap Projections (https://snapprojections.com), - NaviPlan (https://investcloud.com/naviplan-products/). While these platforms are robust, they often suffer from dated user interfaces, complex enterprise pricing, and a lack of modern, integrated Generative AI features. This project aims to disrupt the market by building a lightweight, "AI-native" alternative that prioritizes UX and speed—using the modern stack that tomorrow's developers need to know. The Student Experience (Mentorship Focus): This is a high-fidelity project. Students will be mentored by an experienced product owner and expected to adopt industry-standard DevOps practices. You will gain deep experience with Relational Database design (complex schemas involving beneficiaries, assets, and liabilities) and modern Type-Safe development. Success Metrics: - Deployment of v2.0 to Vercel with full CI/CD pipelines. - Successful migration or re-implementation of the v1 Drizzle schema. - A polished, responsive UI using Shadcn/ui that simplifies complex data entry.

Programming Language(s):
Framework: Next.js 14+ (App Router) Library: React Language: TypeScript (Strict Mode) Database/ORM: PostgreSQL (Neon) via Drizzle ORM Auth: Kinde Authentication UI Components: Shadcn/ui, Tailwind CSS, Recharts Services: Stripe (Payments), UploadThing (File Storage), OpenAI API (AI features) Version Control & DevOps: Github

Hardware/Software Requirements:
Students must provide their own development laptops. Access to GitHub. Familiarity with the above tech stack (or equivalent) Sponsor will provide environment variables and API keys for all paid services (Neon, Stripe, OpenAI, Kinde). Sponsor will provide a GitHub repository invite for this project.

Current Work/Arrangement:
A comprehensive v1 prototype currently exists. - Built on Next.js 14, React, Typescript hosted on Vercel. - Using Drizzle & Neon for relational DB. Uploadthing for file storage. - Shadcn & Recharts for UI. - Stripe, OpenAI, and Kinde as 3rd party services. - Github for version control and CI/CD DevOps. It contains the core business logic and database schema but requires re-architecting for scalability, better UI/UX flow, and codebase cleanup. Advisors currently rely on similar software or manual spreadsheets. The student team will have full access to the v1 GitHub repository to use as a "living specification" for building the refined v2.0.

Previous Project?:
no

Project ID:
1158

Company:
Iron52

Project Areas:
Mobile Development

Company Profile:
We specialize in people meeting people. People ACTUALLY meeting people. For events that are romantic, platonic, or open-ended.

Project Description:
This project has already begun. It is an iOS app whose interface is identical almost to that of the Tinder app. There are udemy.com tutorials that can guide the entire completion of the app-building process. The difference between Tinder and AceOfDates is that Tinder matches on looks, age, and romance...AceOfDates matches based on the following: the user enters a date and time (e.g. Thursday Nov 28 2025 PM) and a dollar amount (e.g. $13) and nearby matches will appear based on event offerings within that budget, for that date. Examples: - Jason is looking for a chess partner Thursday Nov 28 2025 PM for $0 - Michele is looking for a running partner Thursday Nov 28 2025 PM for $0 - John and Sue are looking for a Dungeons and Dragons partner Thursday Nov 28 2025 PM for $0 - Alby is looking for someone to go to a film festival with Thursday Nov 28 2025 PM for $10 i.e. Users swipe on event offerings, not on the person

Programming Language(s):
Swift, SwiftUI

Hardware/Software Requirements:
MAC (or else we can rent macs in the cloud for the students). XCode (or else we can rent macs in the cloud for the students).

Current Work/Arrangement:
The project is paused for development until ISSP.

Previous Project?:
yes

Project ID:
1165

Company:
WorkSafeBC

Project Areas:
Web Development

Company Profile:
WorkSafeBC is a provincial agency based in British Columbia dedicated to promoting safe and healthy workplaces. Established in 1917, it administers the workers' compensation system and partners with employers and workers to prevent workplace injuries, illnesses, and fatalities. The organization provides education, prevention programs, and regulatory compliance support, while offering no-fault insurance coverage for employers and compensation for injured workers. Serving over 2.7 million workers and approximately 285,000 employers, WorkSafeBC also supports recovery, rehabilitation, and safe return-to-work initiatives, ensuring long-term sustainability of workplace safety across the province.

Project Description:
Organizations that apply to become WorkSafeBC approved training providers often struggle to align their programs with regulatory requirements and core competencies. This creates delays, missing information, and repeated back and forth between applicants and reviewers. EduAlign AI is a proposed solution that helps applicants prepare stronger submissions while also supporting WorkSafeBC reviewers who evaluate these programs. This student project focuses on building a system that guides applicants through the key steps needed to submit a complete training provider package. Students would design features that help users map their learning materials to required competencies, check regulatory references for accuracy, confirm accessibility requirements, and document their complaints handling processes. The system could also use simple automation to flag missing information or inconsistencies and help applicants understand what still needs to be completed before they submit. For reviewers, the project could include an organized dashboard or workspace that displays each section of the submission in a clear, consistent format. This makes it easier for certification specialists and subject matter experts to evaluate programs, find information quickly, reduce administrative errors, and focus more on complex decisions rather than repetitive checks. Possible features include: * A guided workflow that walks applicants through each required section * A learning material mapping tool to connect content to core competencies * A regulatory reference checker that helps applicants ensure accuracy * Prompts for accessibility and complaints handling requirements * Automatic flags for missing or incomplete information * A simple reviewer view that organizes content clearly for evaluation * Export or download options for finalized submission packages The domain is training provider certification in workplace health and safety, but we will explain all requirements in plain language. The scope is flexible and can be shaped to match the skills and interests of the student team. WorkSafeBC staff with technical and domain knowledge will be available to support the project, answer questions, and help refine goals throughout the term.

Programming Language(s):
We are open and flexible to using the best technology for the project. Our environment uses Microsoft .NET and Microsoft Azure, but students may choose the tools they are most comfortable with. Options such as Python, JavaScript, React, or other modern development frameworks are acceptable.

Hardware/Software Requirements:
No special hardware is required. Standard laptops or desktops are sufficient. We cannot provide access to our Azure environment. Students may use their preferred development tools and platforms. We will provide guidance on the process, requirements, and general technical considerations.

Current Work/Arrangement:
Not applicable.

Previous Project?:
no

Project ID:
1166

Company:
Applied Research - BCIT

Project Areas:
empty

Company Profile:
BCIT's Center for IoT helps prepare students for the opportunities provided by IoT & AI. The collection and analysis of data goes beyond the realm of information technology and into the world of physical AI

Project Description:
The goal is to develop a mobile application that will be able to provide clear indication of the quality of air in a public restroom and will will provide alerts and recommendation to facilitate an optimized cleaning routine for large organizations such as YVR airport. The mobile device will need to connect by Wi-Fi or Bluetooth to a device that collects parameters such as: CO2, TVOC, temperature, humidity ,PM1.0, PM2.5, PM10 and HCHO (Will be provided by me). The collected data will need to be analyzed and compared to a wide set of conditions to provide recommendations and alerts.

Programming Language(s):
The students may choose their preference.

Hardware/Software Requirements:
HW will be provided by CARI. The final device will be selected with the students to ensure: 1. Meet the required parameter set 2. Provide easy connectivity to a mobile (Wi-Fi or Bluetooth) 3. The device has a clear API that enables development and integration. For example : http://bit.ly/48DXKb9

Current Work/Arrangement:
Currently YVR and most large organizations clean the restrooms on a pre-defined schedule and have no information whether they are cleaning too often or not often enough. They also have no ability to respond to a occurring situation.

Previous Project?:
no

Project ID:
1184

Company:
South Arm Technology Services

Project Areas:
DevOps,Web Development,Scripting/IT

Company Profile:
A Vancouver based technology company that provides IT services and custom application development. On IT services side, focus is cybersecurity; on custom application side, focus is AI and automation.

Project Description:
Peer-Review Recruiting Platform Project Description: A web-based recruiting platform where job applicants review and rate each other using proven comparative judgment methodologies. Instead of traditional recruiter screening, applicants compare pairs of anonymized peer applications (e.g., "Which candidate is stronger: A or B?"). The platform aggregates these pairwise judgments into a ranked shortlist for employers. AI and automation is built in. But in this platform, AI/Automation works as assistant and do not make decisions. This is the core philosophy of this system: put people first. Business Problem: Hiring is broken. Many recruiters lack domain expertise to evaluate candidates, yet charge 15-20% of first-year salary. Meanwhile, AI tools on both sides (resume writers and screeners) have turned hiring into AI-vs-AI, removing human judgment. This platform puts domain peers—who can spot inflated resumes better than recruiters, at the center of screening. It reduces hiring costs, gives applicants transparency and feedback, and produces higher-signal candidate rankings. The unique part of the platform is it gives job applicants more power in the hiring process. Hiring process is a black box to job applicants, and it creates anxiety and feeling of being ignored. This platform is meant to change this. Key Components: Applicant submission portal Anonymization layer (strips PII before peer review) Smart resume formatting Comparative judgment interface (reviewers see pairs, never themselves) AI assistant that prompts reviewers for reasoning when judgments seem thin or inconsistent Employer dashboard showing ranked candidates with summarized peer rationale Reward system: applicants who complete more reviews gain more visibility in the review pool Anchor/Bait system: inject fake resumes to evaluate reviewers' work Expectations for Students: Be familiar with coding, Python or JavaScript. Have an AI-first mindset in development.

Programming Language(s):
Frontend: React Backend: Node.js or Python FastAPI Database: PostgreSQL AI integration: Vendor agnostic Hosting: Cloud-based The back end stack is not decided yet. Currently it's Python, but may entirely move to Node.js.

Hardware/Software Requirements:
Hardware can be provided upon request. Students are welcome to use own laptop. Azure Virtual Desktop (AVD) will be provided. Students will use company identity to login to AVD. Coding tool will be Visual Studio Code. Collaboration platform is Github Enterprise. All necessary software subscriptions will be provided.

Current Work/Arrangement:
This project is currently in active early planning and development stage

Previous Project?:
no

Project ID:
1187

Company:
Patrick Guichon (BCIT)

Project Areas:
Web Development

Company Profile:
I am an instructor at BCIT

Project Description:
There are a number of interactive SQL tutorial sites such as: https://sqlbolt.com, https://sqlzoo.net, https://www.interviewbit.com/problems/study-selection/ and https://sqlzap.com These sites show how to use SELECT statements in SQL Databases and have interactive hands-on exercises where users can practice writing queries, see the results and verify if their results match the required answer for the current question. As an instructor at BCIT and teaching into a number of introductory and advanced database courses, I see the importance for novice students to practice writing select queries in order to master this skill. While I encourage students in my courses to use these online tools, if I make them optional, students will often overlook them and not get the practical hands on skills that they need. The problem with the these above mentioned tools, is that I have no way to know if/when students completed them. I would like to make exercises required as part of the courses I teach and have students submit their work, but it is virtually impossible to confirm that the students have (truthfully) completed the work. My suggested solution and project project proposal for ISSP, is to recreate a similar site for which I can control the login information to help verify that students in my class have completed the work. The site should allow for the creation of lesson modules which group SELECT practice questions into groups based on syntax and difficulty and would log student student interactions. Students would log in, complete their required homework and submit proof of completion. The site needs to be secure so that students (users of the site) are not able to permanently destroy the data within the database (ex: via SQL Injection attack). It needs to also needs to prevent students cheating by claiming they finished the exercises on the site when they didn't. As part of the project, a test script (unit test) will be needed to created.

Programming Language(s):
Bun, Hono, Kinde, MySQL, Drizzle, Typescript, React, Vite,

Hardware/Software Requirements:
IDE such as VS Code Jira for Project/Time/Bug Management GitHub for Version Control

Current Work/Arrangement:
This is the second time this project has been worked on by students. Students are expected to work with an existing code base.

Previous Project?:
yes

Project ID:
1189

Company:
BAM Robotics

Project Areas:
empty

Company Profile:
BAM (Break & Make) Robotics is a Vancouver start-up developing low-cost AI-powered robots that sort mixed waste with the dexterity of a human and the speed and reliability of a machine. Pilot applications include sorting medical waste with a BC medical waste aggregator. The long-term vision is to develop a modular robotics system for the circular economy that can execute both "making" (including assembly) and various kinds of "breaking" (including sorting and disassembly.) Co-Founder & CEO/CTO, Zach Yamaoka: - Product Team @ Automata (UK start-up with $60M raised to develop low-cost robot arm) - NPI Robotics Engineer @ Dyson - Founder of 2 x successful Medical Technology Companies (Ambience PPE & D.future) - Audi Autonomous Car Cup Software Engineer @ Imperial College Robot Intelligence Lab Co-Founder & COO, Anna Yamaoka: - New York attorney - Researcher @ Urbanlogiq and Guarnini Global Law & Tech - 1st Class Honours and IP Law Prize from Oxford University (Law)

Project Description:
Accurate object detection in recycling and waste facilities is an open challenge. Typical issues include cluttered scenes, partial occlusions, dirt and contamination, object deformation (crushed, folded, torn), visually similar materials (e.g., plastic vs glass, or different plastics that look the same.) Hyperspectral imaging aims to provide more reliable object detection by capturing the scene across a wide spectrum of wavelengths, not just RGB. While objects may look very similar in just the 3 RGB channels, they often have distinct spectral signatures across many wavelengths. Because hyperspectral features are tied to material chemistry, they are much more robust to clutter, occlusion, deformation. (E.g. no matter how you crush a PET bottle, its spectral signature still indicates that it's plastic.) A hyperspectral camera by itself does not output "material =PET" or "object = bottle". It simply returns intensity values at each wavelength. Supervised ML models are then trained to map these spectral signatures to target labels (material, object type, etc.). The major drawback of hyperspectral cameras is cost. For short-wave infrared (SWIR) hyperspectral cameras, we received quotes in the $20k–$50k range. In our design, we use many small robot arms, each doing ~60 picks per minute over a conveyor belt. To match the throughput of a large $200k optical sorter, we can deploy ~6 small arms costing around $20k in total. With this distributed approach, each small arm gets its own camera. The challenge is that it would be prohibitively expensive to give each arm its own hyperspectral camera, and runs counter to our goal of producing a low-cost robotic solution. So our goal is to match the performance of a hyperspectral system using only cheaper RGB cameras. In other words: How can we predict the same outputs as the hyperspectral camera (material, product type, and per-pixel instance) using only 3 RGB channels? Our idea: Use the hyperspectral camera as a "superior knowledge"oracle. Train a model that predicts the same labels using only RGB, using the hyperspectral model's predictions as supervision. We are acquiring a hyperspectral and RGB Camera. We are taking as a starting point the Spectral Waste paper and Data Set: https://arxiv.org/abs/2403.18033 STUDENT CHALLENGES: There are several related sub-problems and datasets we want to work with. #1 - RGB + Spectral → Material/Object/Instance Spectral Waste Paper and baseline models: GitHub: https://github.com/ferpb/spectralwaste-segmentation Kaggle: https://www.kaggle.com/datasets/vijaynyayavn/spectralwaste Goal: Reproduce the Spectral Waste paper's RGB +HYPER results. #2 - RGB → Material/Object/Instance #2(i) Spectral Waste Dataset We reuse the SpectralWaste dataset:- https://github.com/ferpb/spectralwaste-segmentation …but this time we only use the RGB images Goal: Reproduce paper results for RGB #2(ii) Zero Waste Dataset Downloads from here: Zero waste dataset: https://ai.bu.edu/zerowaste/ Zero waste v2 dataset: https://github.com/dbash/visda2022-org Goal: Reproduce paper results #2(iii) Zero Waste Dataset + Spectral Waste Goal: Combine both datasets and try to achieve a superior result An interesting challenge is that the spectral RGB is a line scanner while the zero waste is an area scanner (this is described in notes below) #3 - RGB → Spectral #3(i) NTIRE 2022 Spectral Recovery Challenge Overview paper : https://openaccess.thecvf.com/content/CVPR2022W/NTIRE/papers/Arad_NTIRE_2022_Spectral_Recovery_Challenge_and_Data_Set_CVPRW_2022_ Data Download: https://github.com/boazarad/ARAD_1K Winner github repo here with pytorch model: https://github.com/caiyuanhao1998/MST-plus-plus?tab=readme-ov-file Goal: Reproduce winning paper #3(ii) Spectral Waste Dataset We again reuse SpectralWaste: https://github.com/ferpb/spectralwaste-segmentation But now we change how we use the data. Original setup: Inputs:256 ×256 ×3 RGB 256 ×256 ×224 spectral (aligned to RGB) New setup: Inputs:256 ×256 ×3 RGB Outputs:256 ×256 ×224 spectral (aligned to RGB) So we train a model to predict the full spectral cube from RGB alone. This is essentially within-image self-supervised learning: "Predict the occluded from the visible"– analogous to masked language modeling for LLMs Key points: No manual labeling is required. You can simply run a hyperspectral +RGB setup above a conveyor and record data continuously. Limitation: This does not directly give us material / object type / instance labels. However, it follows the GPT-style paradigm:Self-supervised pretraining (RGB →spectra). This is then supervised fine-tuning for material / object / instance prediction Goal: Phase 1 –Pretraining within the image Inputs:256 ×256 ×3 RGB Outputs (prediction):256 ×256 ×224 spectral (predicted) Labels (ground truth):256 ×256 ×224 spectral (measured) Phase 2 –Downstream task Use the pretrained features to try to beat the RGB-only benchmark on material/object/instance prediction. Pretraining allows us to use the entire multimodal dataset (7,655 RGB +spectral image pairs), not just the 852 labeled examples #3(iii) Spectral Waste Dataset Zero Waste + Spectral Waste Phase 1 -Pretraining within image (Spectral Waste) Phase 2 - Training on labels (Spectral Waste) Phase 3 -Training on labels (Zero Waste) This represents the entire workflow. The only difference is the zero waste labels here are provided by humans but in our deployment would be provided by the hyperspectral oracle Goal: Try to achieve the highest score of any model

Programming Language(s):
Python.

Hardware/Software Requirements:
Laptop/personal computer.

Current Work/Arrangement:

Zach Yamaoka is in charge of technical development (hardware and software) and would work directly with students.

Previous Project?:

no

Project ID:
1192

Company:
Center for Applied Second Language Studies / University of Oregon

Project Areas:
Web Development,Mobile Development,Games,AR/VR,Machine Learning/Algorithms/Research

Company Profile:
The Center for Applied Second Language Studies (CASLS) at the University of Oregonsupports innovative world language teaching and learning. We develop research-based curricula, professional development resources, and assessments to support infrastructure and innovation. We are one of sixteen National Foreign Language Resource Centers that work to increase the nation’s capacity for language education. Our resources incorporate best practices in pedagogy, the latest in second language acquisition research, and technology that’s actually usable in the classroom.

Project Description:
The VAuLT platform is a no-code mixed-reality toolkit developed at the Center for Applied Second Language Studies (CASLS). It allows users to build place-based, interactive experiences—escape rooms, scavenger hunts, cultural site activations, and other narrative-driven XR interactions—without advanced technical skills. What the Project Is Doing: We are upgrading VAuLT by developing a new cross-platform front-end (Unity or web-based) that connects to the existing back-end. This will make the system more accessible, more flexible, and better suited for educational and community storytelling projects. Students involved in this project will: Build and test components of the new front-end interface. Integrate front-end prototypes with an already-functioning back-end. Strengthen UI/UX, navigation, and interaction design for mixed-reality experiences. Participate in short design cycles to refine functionality through hands-on testing. What Makes VAuLT Interesting Active back-end + real use cases: You are not starting from scratch; you’re extending a working system. Mixed-reality design without heavy asset pipelines: The platform uses templates rather than complex 3D modeling, so engineering work focuses on interaction logic, not graphics. Immediate impact: Your work directly affects how educators, students, and community partners build XR experiences.

Programming Language(s):
Swift, PHP, HTML & CSS, JavaScript

Hardware/Software Requirements:
iPhone running iOS 18 or newer. Web browser

Current Work/Arrangement:
Student involvement will focus on advancing the process by prototyping both a Unity-based and a web-based front end, testing their connection to the existing back-end, and helping move the project from stalled prototypes into an active development cycle.

Previous Project?:
no

Project ID:

1200

Company:

Client: Conscious Connections Network (Inc. Pending)

Project Areas:

DevOps,Web Development,Mobile Development

Company Profile:

Conscious Connections Network (under Agora Network Technologies Inc.) builds technology that supports healthier, more intentional relationships. Our products combine modern psychology, AI-assisted communication tools, and elegant app design to help individuals and couples understand themselves, communicate better, and form meaningful connections. Agora, our parent organization, has sponsored 10+ successful BCIT ISSP projects across AI, recommender systems, and full-stack app development. Conscious Connections expands this work into the dating and relationship wellness space through tools like compatibility algorithms, guided communication experiences, and soon, an AI-supported conflict resolution platform. Our mission is to create emotionally intelligent technology that strengthens human connection, promotes self-awareness, and empowers people to build relationships with depth, intention, and compassion.

Project Description:

Notion Doc: https://fallacious-ozraraptor-f59.notion.site/AI-Assisted-Conflict-Resolution-Tool-2c2ee584d2ee80a59701c6948350232d?source=copy_link **Client:** Conscious Connections Network (Inc. Pending) *(owner of Agora Network Technologies Inc.)* **Project:** Conscious Connections: AI-Assisted Conflict Resolution Tool (Web + Mobile Monorepo) --- ### **Client** **Conscious Connections** is a values-based dating and relationship platform focused on intentional, emotionally intelligent partnership. It sits under the broader **Agora Network Technologies** umbrella, which has sponsored 10+ successful BCIT ISSP projects across recommender systems, computer vision, and app feature development. Conscious Connections already has: - A working **Next.js/TypeScript** codebase. - A v1 **matching algorithm** and survey system (previous ISSP term). - A clear product vision around **healthy relationships**, not just swiping and casual dating. The next major evolution is to support **existing couples** (and early-stage relationships) with tools that improve **communication, conflict resolution, and nervous system safety**—using AI as a coach, translator, and gentle facilitator. This ISSP project will design and build the first version of the **Conscious Connections Conflict Resolution Tool**: a shared, AI-assisted chat space that helps couples fight less destructively, understand each other more deeply, and de-escalate conflict through guided communication. --- ### **Project Title** **Conscious Connections: AI-Assisted Conflict Resolution Tool (Web + Mobile Monorepo)** --- ### **Overview** Most couples never learned how to fight well. Arguments spiral, old wounds get triggered, and the conversation becomes about *winning* instead of understanding. Conscious Connections wants to change that by giving couples a **shared space**—think “therapist in the chat”—where: - Each partner can express themselves honestly. - An AI “therapist” helps **clean up** heated messages before they’re sent. - The system can gently **call out patterns**, suggest more constructive language, and highlight emotional needs in real time or on demand. This ISSP term is focused on building the **core technical foundation**: 1. A **Turborepo-style monorepo** with: - **Expo React Native app** (primary client). - **Next.js web app** (secondary client + API server). - Shared **TypeScript models, UI primitives, and OpenAI utilities**. 2. A secure, scalable **authentication and subscription system** using **Clerk** for auth (with the understanding that there may be an ongoing discussion with the client/program about long-term auth strategy) and **Stripe** for couple-based billing. 3. A **real-time chat system** (WebSockets) where two partners can converse in a shared “relationship room.” 4. A structured **intake and history system** that builds a psychological profile for each user and relationship—forming the foundation for AI context and coaching. 5. Initial **AI-assisted conflict resolution features**: - “Clean My Message” (rewrite heated text into compassionate, clear language). - On-demand **therapist-style reflections** on a conversation thread. The term is designed into **4–5 main modules**, with one or two students owning a core “thread” while collaborating at integration points. --- ### **Your Opportunity** This project sits at the intersection of: - **Full-stack engineering** (Next.js, Expo, monorepo architecture, WebSockets). - **Applied AI** (OpenAI integration, prompt design, context management). - **Relationship psychology** (communication patterns, conflict repair, emotional safety). As a member of this ISSP team, you will: - Set up a **real monorepo** that supports *both* web and mobile apps. - Implement **real-time communication** with persistent message history. - Work with **auth, subscriptions, and partner linking** (two accounts sharing one “relationship space”). - Build the first version of tools that could genuinely **help couples fight better**. - Ship **polished MVP features**, not just prototypes, under the guidance of an experienced founder/CTO. This is production-oriented work: your code will form the foundation of a product that Conscious Connections intends to put into people’s hands. --- ### **Project Goals** **Objective:** Deliver a **polished MVP** of the Conscious Connections Conflict Resolution Tool, built on a **Turborepo monorepo** (Expo + Next.js), with: - Auth, onboarding, and “relationship room” creation. - Structured psychological intake for each partner. - Real-time chat with persistent history. - Initial AI conflict support tools (message clean-up + on-demand reflections). - Subscription logic for couple-based billing. Each student will be assigned a **primary thread** after the opening meeting, based on interests and strengths. Threads are designed to be modular but interconnected. --- ### **Thread 1: Monorepo Architecture & Core Infrastructure** **Goal:** Establish a stable, scalable **monorepo** that hosts both the web and mobile clients plus shared libraries. **Responsibilities:** - Set up a **Turborepo** (or similar) with: - `apps/web` → **Next.js** (web app + API routes). - `apps/mobile` → **Expo React Native** app. - `packages/ui` → Shared components (where feasible). - `packages/core` → Shared types, validation schemas, and domain logic. - `packages/api-client` → Shared API clients/hooks. - Configure: - TypeScript project references. - ESLint/Prettier. - Basic testing setup (e.g., Jest or Vitest). - Environment variable handling across apps. - Ensure developers can: - Run web + mobile concurrently in dev. - Share interfaces and models across both clients. - Consume the same backend (`Next.js` API routes). **Deliverables:** - Monorepo with documented structure and setup instructions. - Dev scripts for running: - `web` (Next.js dev server). - `mobile` (Expo dev). - Shared types for: - `User`, `Relationship`, `Message`, `IntakeQuestion`, `IntakeAnswer`, etc. - Basic CI workflow (lint + tests) for GitHub. --- ### **Thread 2: Authentication, Onboarding & Relationship Linking** **Goal:** Implement secure **user authentication**, **onboarding flows**, and **relationship linking**—allowing two separate user accounts to share one conflict-resolution “room.” **Responsibilities:** - Integrate **Clerk** for auth across: - Next.js web app. - Expo React Native app. *(With the understanding that there may be an ongoing discussion with the client/program about the long-term auth provider; implementation should be modular and well-structured.)* - Implement user onboarding flows: - Account creation (email or

OAuth, depending on configuration). - Basic profile setup (name, pronouns, relationship status, etc.). - Implement “relationship workspace” mechanics: - User A creates a **new relationship** and invites User B via email. - User B accepts invite, creates/logs into account. - Both accounts are then linked to a shared `Relationship` record. - Define subscription logic using **Stripe**: - At MVP: one subscription per relationship. - Support flows like: - One partner pays for both. - Ability to later support split billing (design schema; full UX may be a stretch goal). **Deliverables:** - Auth working in both web and mobile clients. - Relationship model with: - `Relationship`, `RelationshipMembership` tables (Prisma). - Invite/join flow for partners (at least implemented on web; mobile is a plus). - Basic subscription check (e.g., `relationship.isActive`) enforced before chat features are enabled. --- ### **Thread 3: Intake & Psychological History System** **Goal:** Create a **structured intake system** that gathers key psychological and relational background for each user and relationship—forming the base “memory” that AI can draw from later. **Responsibilities:** - Design a flexible **JSON-driven question schema**: - Supports question types like: - Text input. - Multi-select / checklists. - Themes include: - Early childhood dynamics (caregivers, emotional climate). - Socioeconomic background. - Experiences like bullying, eating disorders, adoption, etc. - High-level attachment or conflict patterns (non-diagnostic language). - Implement intake flows in: - Web app (primary). - Mobile app (basic version, if time permits). - Store responses in Prisma models such as: - `UserIntakeProfile` - `IntakeAnswer` (per-question answers). - Implement a function to compile a user’s intake into a single **summary object** or text block that can later be used as AI context (e.g., in an OpenAI “profile” file or internal long-term context store). **Deliverables:** - Prisma schema and migrations for intake data. - Frontend UI for completing intake (web, mobile if feasible). - Validations and basic UX safeguards (save progress, etc.). - Documentation on how this data will be used by later AI components. --- ### **Thread 4: Real-Time Chat & AI-Assisted Conflict Tools** **Goal:** Build the **real-time chat experience** between two partners and integrate initial **AI conflict resolution tools**. **Responsibilities:** - Implement a **conversation model**: - `Message` table with: - `relationshipId` - `senderUserId` - `content` - `createdAt` - `messageType` (e.g., `user`, `ai_reflection`, `ai_rewrite_suggestion`). - Implement **real-time messaging** using WebSockets: - Likely via: - Vercel’s WebSocket support, or - Socket.IO, or - Pusher (depending on constraints). - Ensure: - New messages appear in real-time for both partners. - Messages are persisted to the DB. - Build the chat UI: - Shared room view for a relationship. - Message list, input bar, typing indicator (optional stretch). - Implement initial AI tools using OpenAI: 1. **“Clean My Message” Tool** - User types a raw, emotional message. - AI rewrites it into: - Non-accusatory - Specific, emotionally honest - Non-hostile, but still clear. - User can then send the AI-refined version. 2. **On-Demand Therapist Reflection** - User or couple taps a button like “Get Reflection.” - Backend sends recent segment of conversation + high-level intake summary to OpenAI. - Returns: - Observations about patterns. - Gentle coaching (what each partner may be feeling, suggestions for repair attempts, etc.). - AI appears as a third voice (“Therapist” or “Guide”) in the chat. - Ensure **safety and clarity**: - Make it visually obvious which messages are AI-generated. - Do *not* let AI speak *as* one of the partners. **Deliverables:** - Web-based chat UI working end-to-end (real-time + persistence). - Basic AI endpoints: - `/api/ai/clean-message` - `/api/ai/reflection` - At least one “happy-path” flow demonstrated: - Two logged-in users in the same relationship. - They chat. - One uses “Clean My Message.” - They request an AI reflection. --- ### **Stretch Goals (If Time Allows)** - Therapist “live in the room” mode where AI periodically interjects automatically (e.g., every X messages). - More sophisticated context management for AI (e.g., conversation summarization and memory). - Mobile parity for all major features (full chat + intake on Expo). - Expanded Stripe flows (split billing, plan tiers). --- ### **Technology Stack** - **Frontend / Clients:** - **Next.js** (TypeScript) for web app and API routes. - **Expo React Native** for mobile client. - **Tailwind CSS** (web) and shared design tokens where possible. - **Monorepo & Tooling:** - **Turborepo** (or similar) for monorepo management. - TypeScript project references, ESLint, Prettier, Jest/Vitest. - **Auth & Identity:** - **Clerk** for authentication and session management across web and mobile, with the understanding that there may be an ongoing discussion about auth provider choice long-term. - **Backend & Data:** - Next.js **API routes** for REST endpoints and AI features. - **Prisma** ORM with **PostgreSQL**. - **Real-Time Messaging:** - WebSockets via a suitable service (e.g., Vercel WebSockets, Pusher, or Socket.IO) integrated with Next.js. - **Payments:** - **Stripe** for subscriptions (relationship-level billing). - **AI:** - **OpenAI API** for: - Message rewriting / tone transformation. - Therapist-style reflections. - **DevOps & Hosting:** - Web: **Vercel** for Next.js. - Mobile: Expo workflow. - DB & ancillary services: existing Agora/Conscious Connections infrastructure (Postgres on cloud). --- ### **Mentorship & Support** You’ll be working directly with: - **Taylor Aucoin** – Founder & CTO, Conscious Connections / Agora Network Technologies - Role: **Product & Technical Lead** - Responsibilities: - Define product behaviors and acceptance criteria. - Provide architecture guidance (monorepo layout, DB modeling, AI integration patterns). - Weekly check-ins and code review support. - Write ClickUp-style tickets and implementation notes. You will also have: - Access to **existing Conscious Connections codebases** (Next.js, Prisma, etc.). - Prior ISSP documentation and patterns used successfully in Agora projects. - Slack/Discord channel for async questions and guidance. --- ### **Expectations for Students** - **Prerequisites:** - Comfortable with **JavaScript/TypeScript** and **React**. - Some familiarity with backend concepts (APIs, databases). - Interest in working with AI APIs is a plus. - **Ownership:** - Each student will own a **primary thread** (see above) but is expected to collaborate where threads intersect. - You’ll be responsible for your part from **design to implementation to documentation**. - **Collaboration & Communication:** - Weekly calls for status updates, questions, and architecture decisions. - Active communication in Slack: ask for help early, surface blockers, share progress. - **Quality & Delivery:** - Code should be: - Typed (TypeScript). - Reasonably tested (happy-path tests at minimum). - Documented (README updates, inline comments where logic is non-obvious). - Target: **polished MVP**, deployable and usable in a test/staging environment by end of term. --- ### **Closing Statement** This project is about more than building a chat app. It’s about giving couples a safer way to handle some of the hardest moments in their relationship—when they’re hurt, triggered, or afraid. By combining **solid engineering**, **clean architecture**, and **thoughtful AI design**, you’ll help build a tool that can actually reduce harm and increase understanding between real people. You’ll leave with experience in: - Monorepos and multi-platform architecture. - Real-time systems and WebSockets. - Auth + subscription flows in a production-style environment. - AI-assisted UX for emotionally sensitive contexts. If you’re excited to build something technically interesting **and** deeply human, this is your chance to help shape the first generation of AI-supported conflict resolution for conscious relationships.

Programming Language(s):

The project will primarily use TypeScript across all components. The web application and backend API will be built with Next.js (TypeScript), while the mobile application will use React Native via Expo (TypeScript). Additional technologies include Prisma (TypeScript) for database access and WebSocket libraries for real-time communication. If AI integration is implemented, it will use TypeScript-based server endpoints that interact with the OpenAI API.

Hardware/Software Requirements:

Students will need access to a standard development laptop capable of running modern JavaScript tooling (Node.js, TypeScript, Expo).

No specialized hardware is required. Software requirements include: • Node.js (LTS version) • TypeScript • Next.js for the web application and backend API • Expo / React Native for the mobile app • Prisma ORM and access to a PostgreSQL database • GitHub for version control • Vercel CLI (optional, for deployment) • Stripe developer tools (for subscription integration) • Clerk developer account (authentication) • OpenAI API access (for AI-assisted features)

Current Work/Arrangement:

NA

Previous Project?:

no

Project ID:
1203

Company:
South Arm Technology Services

Project Areas:
Scripting/IT

Company Profile:
South Arm Technology is an IT services company providing cybersecurity, IT management, and business automation services. On IT services side, the focus is to script and automate as much as possible to reduce costs and improve accuracy and efficiency.

Project Description:
1. Executive Summary The objective of this project is to architect and deploy a low-cost, secure, and highly efficient IT management ecosystem. By moving away from traditional, proprietary agent-based Remote Monitoring and Management (RMM) platforms—which often present significant supply-chain security risks—this project utilizes a "secure-by-design" philosophy. The ecosystem will be built primarily on open-source protocols and tools, prioritizing lightweight connectivity, granular observability, and automated security compliance. 2. Project Scope & Technical Components The project is divided into seven core technical pillars: A. Secure Connectivity (Mesh Architecture) Objective: Eliminate reliance on heavy, insecure RMM agents for connectivity. Approach: Deploy a WireGuard-based mesh VPN (e.g., Netbird or Tailscale) across all client endpoints. Technical Detail: The VPN will strictly provide the transport layer (connectivity) without handling authentication/login application layers, ensuring a clear separation of concerns. This establishes a secure, encrypted tunnel to endpoints without the vulnerability surface area of traditional RMM agents. B. Observability & Monitoring Objective: Implement comprehensive system visibility without performance degradation. Approach: Utilize a modern, open-source observability stack focusing on agentless or lightweight-agent architectures. Tooling: Zabbix: For infrastructure status and performance metrics. Vector.dev: For high-performance log and metric aggregation. Graylog: For centralized log management and analysis. C. Patch Management & Software Distribution Objective: Streamline Windows software management and security updates. Approach: Leverage Winget (Windows Package Manager) as the primary engine for software installation and patching. Technical Detail: To ensure package integrity and version control, the team will build and host a Private Winget Repository, allowing for curated and approved software deployment rather than relying solely on public repositories. D. Vulnerability Management Objective: Proactive detection of security risks and CVEs. Approach: Integrate continuous vulnerability scanning into the management ecosystem. Tooling: OpenVAS (Greenbone) will be deployed to regularly scan network endpoints and identify vulnerabilities, outdated configurations, or unpatched services. E. Automation & Orchestration Objective: Achieve a state of "Infrastructure as Code" (IaC) to eliminate manual configuration drift and reduce administrative overhead. Approach: We will combine Ansible for high-level orchestration and configuration management with PowerShell for granular Windows-specific automation. Tooling & Workflow: Ansible: Will serve as the central controller to push "Playbooks" across the WireGuard mesh. It will manage state, enforce configuration standards, and orchestrate complex deployments across multiple endpoints simultaneously. PowerShell: Will be utilized heavily as the execution engine for Windows endpoints. Ansible will trigger PowerShell modules and scripts to handle deep OS-level tasks, registry modifications, and software interactions. F. Remote Support & Access Objective: Provide on-demand remote desktop support while maintaining data sovereignty. Tooling: MeshCentral: For web-based remote management and terminal access. RustDesk: As a high-performance, self-hosted alternative to TeamViewer/AnyDesk. G. Windows Fast Imaging Objective: Enable rapid deployment of new devices and disaster recovery through fast re-imaging services. Approach: Implement a network-based imaging solution that separates the base OS image from driver management to ensure speed and hardware compatibility. Tooling: Microsoft Deployment Toolkit (MDT) & Windows Server: To manage and deploy "Gold" images and task sequences over the network. Vendor-Specific Driver Management: Tools such as Lenovo Vantage (and equivalent commercial Vantage tools for enterprise) will be integrated to automate post-imaging driver updates and firmware management, ensuring hardware is fully optimized immediately after re-imaging.

Programming Language(s):
PowerShell Python

Hardware/Software Requirements:
No specific requirement Azure Virtual Desktop, Github Enterprise, etc, will be provided.

Current Work/Arrangement:
Some components have already used in production environment.

Previous Project?:
no

Project ID:
1206

Company:
British Columbia Institute of Technology - Computing Flexible Learning

Project Areas:
Web Development,Scripting/IT,Other

Company Profile:
The School of Computing and Academic studies: is the largest provincial provider of computing graduates and has the most extensive Part-time Studies computing offering in Western Canada.

Project Description:
Project Title Workflow Automation for Academic Advising and Student Success using Microsoft Power Automate ***Project Description*** This project will design and implement a workflow automation solution using Microsoft Power Automate to streamline academic advising and student success processes within BCIT's Flexible Learning programs. The solution will: - Capture student requests and course planning data via Microsoft Forms. - Store and manage data in SharePoint Lists for structured tracking. - Automate approval workflows for course plans and credential evaluations. - Generate and email standardized documents (e.g., course plans, confirmations) to students and administrators. - Provide real-time notifications in Microsoft Teams for status updates. ***Goals & Expected Outcomes:*** - Reduce manual effort and errors in advising workflows. - Improve turnaround time for student requests. - Deliver a functional Power Automate solution integrated with Microsoft 365 apps (Outlook, Teams, SharePoint). - Provide documentation and a demo for future scalability.

Programming Language(s):
Python, if required

Hardware/Software Requirements:
Microsoft 365 environment (Power Automate, SharePoint, Teams, Outlook). Access to BCIT advising workflows and sample data.

Current Work/Arrangement:
Currently, advising and course planning involve manual email exchanges, document preparation, and status tracking across multiple systems, leading to delays and inefficiencies.

Previous Project?:
no
