Index

- Home

Hacking Materials User Interface

About

Materials science is a complex field that focuses on the study of physical materials and their properties. MatMiner is an existing Python library that combines materials data with Machine Learning strategies and models to study material properties without the need for time-consuming physical experimentation. Our project, Hacking Materials ("HA"), aims to build an easy and intuitive user interface to the MatMiner library* to eliminate the need for its users to have substantial Python or machine learning knowledge or experience.

Materials engineering is a field in which physical materials (e.g. metals, ceramics, polymers, composites, etc.) are studied to understand their composition, characteristics and properties. In many industries, such as mining, manufacturing and others, finding the right material for each job is essential for success. Materials engineers and scientists follow many different methods to compare candidate materials and make recommendations based on how they each satisfy the specific use case requirements. Some of these methods include physical experimentation, which can in some cases take up to 20 years to complete.

To avoid this, computer simulations (based on existing databases of known material properties and machine learning algorithms) can be used to compare materials in a much more efficient way. One such tool for this approach is the Python library MatMiner, which allows easy access to ready-made datasets and integrates well with other machine learning Python libraries. However, to use this library, the user must have a substantial level of specialised knowledge in machine learning and programming, which most materials engineering do not have. To help them overcome this, the solution proposed by the client describes a simple and intuitive user interface that would act as a bridge between the user and the MatMiner library in the backend. The vision of this product is to make machine learning methodologies more accessible within the materials science and engineering industry as a whole, minimising the time and financial costs involved and leading to a more efficient industry.

Contact Details

Quick Links

Slack	Trello	GitHub
link	link	link

Client

Name	Email
Dr Christian Brandl	christian.brandl@unimelb.edu.au

Staff

Name	Email	Role
Mauro Mello Jr	mauro.mellojr@unimelb.edu.au	Supervisor
Eduardo Oliviera	eduardo.oliveira@unimelb.edu.au	Subject Coordinator

Team contacts: **BoxJelly** 🪼

Name	Email	Role	Student ID
Zhaoqi Wang	zhaoqiw@student.unimelb.edu.au	Team Rep & Frontend developer	1018092

Dara O hEidhin	doheidhin@student.unimelb.edu.au	Backend developer	1006587
Felipe Leefu Huang Li	fleefuhuangl@student.unimelb.edu.au	Frontend developer	1202652
Radhimas Djan	djanr@student.unimelb.edu.au	Backend developer	1146240
Yaoming Xuan	yaomingx@student.unimelb.edu.au	Backend developer	1167302

Team contacts: **Redback** 🕷️

Name	Email	Role	Student ID
Alastair Daivis	adaivis@student.unimelb.edu.au	Team Rep	359 101
Chunbaixue Yang	chunbaixue@student.unimelb.edu.au	Scrum Master	1208333
Ghina Yashar	gyashar@student.unimelb.edu.au	Frontend Development Lead	1274878
Mamta Lopes	mlopes@student.unimelb.edu.au	Developer	1157314
Sanjeevani Avasthi	savasthi@student.unimelb.edu.au	Developer	1101265

Team contacts: **Bluering** 🐙

Name	Email	Role	Student ID
Yanan Liu	yananl7@student.unimelb.edu.au	Team Rep	1289747
Hongpeil Lu	hongpeil@student.unimelb.edu.au	Developer	1275238
Jiahao Ju	jiahaoj1@student.unimelb.edu.au	Developer	1128182
Rui Zhang	rzzhan2@student.unimelb.edu.au	Developer	1221568
Xinle Yu	xinley@student.unimelb.edu.au	Developer	1294310

Recent space activity

Zhaoqi Wang

| Set-up SSO Authorisation Guide updated about an hour ago • view change

Sprint 4 - Checklist updated about 2 hours ago • view change

Radhimas Djan

| Meetings updated about 2 hours ago • view change

2022-10-28 - Meeting Minutes created about 2 hours ago

2022-10-15 - Making presentation slide Meeting Minutes created about 2 hours ago

Space contributors

- Zhaoqi Wang (56 minutes ago)
- Radhimas Djan (2 hours ago)
- Felipe Leefu Huang Lin (13 hours ago)
- Yaoming Xuan (20 hours ago)
- Dara O hEidhin (1 day ago)
- ...

Handover Guide

HA-2022-handover.pdf

| Development

| Development Process

- All tests are required to pass in CI(continuous integration) before landing a pull request
    - Continuous integration is the process of automating the integration of codes into a project by testing. It is specifically used in our repositories whenever there's a pull request, a backend test will be tested on the code to make sure there will be no error on the project whenever a code is pull requested
- Sprint lifecycle:
    - Sprint Kickoff:
        - Review and re-estimate tasks: user stories get t-shirt size and priority
    - Development:
        - Feature kickoff:
            - Specify test cases and acceptance criteria
            - Tasks are estimated in the number of days to complete using the magic estimation approach
        - Code reviews:
            - Require tests pass in CI before merging
            - At least one other RedBack member must approve the pull request before it can be merged
            - At least one BoxJelly member working on the same categories(backend / frontend) of the code must approve the pull request before it can be merged. For example, if the member has worked on the front/backend of the project, then he /she can approve the code reviews with the front/backend label on the pull request.
            - At least one BlueRing member working on the same categories(backend / frontend) of the code must approve the pull request before it can be merged. All test cases and acceptance criteria identified in kickoff must be satisfied.
        - Use auto-formatters to maintain code quality
        - Branching
            - Use the format `feature/t-<ticket>` as a feature branch template, where `<ticket>` is the Trello card number
            - `<username>/idea` for scratch / experimenting branches
            - `main` is the main branch
            - We will follow the following guidelines: https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow (we probably only need main, release, scratch, and feature branches)
    - Release management:
        - Deploy as required

** Co developed with Bluerings, and Boxjelly, written up by Redback.*

| Quality Assurance Guidelines

Work quality guidelines
- Code should be able to run, there's no syntax error
- If test(s) are included in the pull request, they should pass
- All previous tests should still pass with the new changes in the pull request
- A branch should not be put up for a pull request if it has merge conflicts with main branch, i.e., all conflicts should be resolved before that
- Code should be understandable and contain code documentation as a comment.
- Code should not include global state

Code review guidelines
- Has to be reviewed by 1 member from the other 2 teams
- Pull request should have assigned reviewers within 24 hours once it's no longer DRAFT and should be reviewed (either approve or require changes) within 48h
- Pull requests should include commit messages describing the work you've done and steps you did to verify your work
- Pull requests will be reviewed by using the Review Changes button under the Files changed tab
- Pull requests should only be merged/rebased by the creator of the pull request
- Branch that has been successfully merged/rebased to main should be deleted by the creator of the pull request
- Reviewers should follow the work quality guideline to review the code

Acceptance criteria definition guidelines
- It should be defined from the user's point of view
- It should contain a list of steps to test the desired functionality which is based on the Given, When, Then format.

Definition of done (for user story)
- Acceptance criteria should be defined for the user story and pass
- All related code has passed code review and merged to main

*Co-developed *with Bluerings and Boxjelly, written up by Redback*

| Frontend workspace structure proposal

Proposed by	Ghina Yashar
Presented to	All frontend contributors from teams RedBack, BlueRing and BoxJelly
Proposal date	September 1 2022
Status	APPROVED - September 2 2022
Approvers	• Mamta Lopes(RedBack): 1/9/2022 • Felipe Lin (BoxJelly): 2/9/2022 • Rui Zhang (BlueRing): 2/9/2022

Proposed Structure

Please note: All the names used below can be replaced if needed. The focus of this proposal is more on the structure rather than the naming.

Summary video

If you don't like reading, please watch the video below for a quick overview of the proposed structure. The sample code snippets shown in the video are copied below as well.

Proposal summary.mov

Summary in writing

The structure I'm proposing would follow this rough directory tree:
Sample directory structure

```
|_assets
|_src
    |_components
        |_exampleComponent
            |_examples.tsx
            |_index.tsx
            |_test.tsx
            |_styled.tsx
        |_ dropdownSelectStepType
            |_examples.tsx
            |_index.tsx
            |_test.tsx
            |_styled.tsx
        ...
    |_steps
        |_datasetSelection // (e.g.)
            |_index.tsx
            |_test.tsx
            |_HelpModal
                |_index.tsx
                |_styled.tsx
                ...
            ...
        ...
    |_sections
        |_appHeader
            |_index.tsx
            ...
        |_appBody
            |_InputPanel
                |_index.tsx
                ...
            |_ViewingWinow
                |_index.tsx
                ...
            |_index.tsx
        |_appFooter // amendment suggested by Felipe
            ...
    |_App.tsx
    ...
|_package.json
|_README.md
...
```

The main ideas of this structure are as follows:

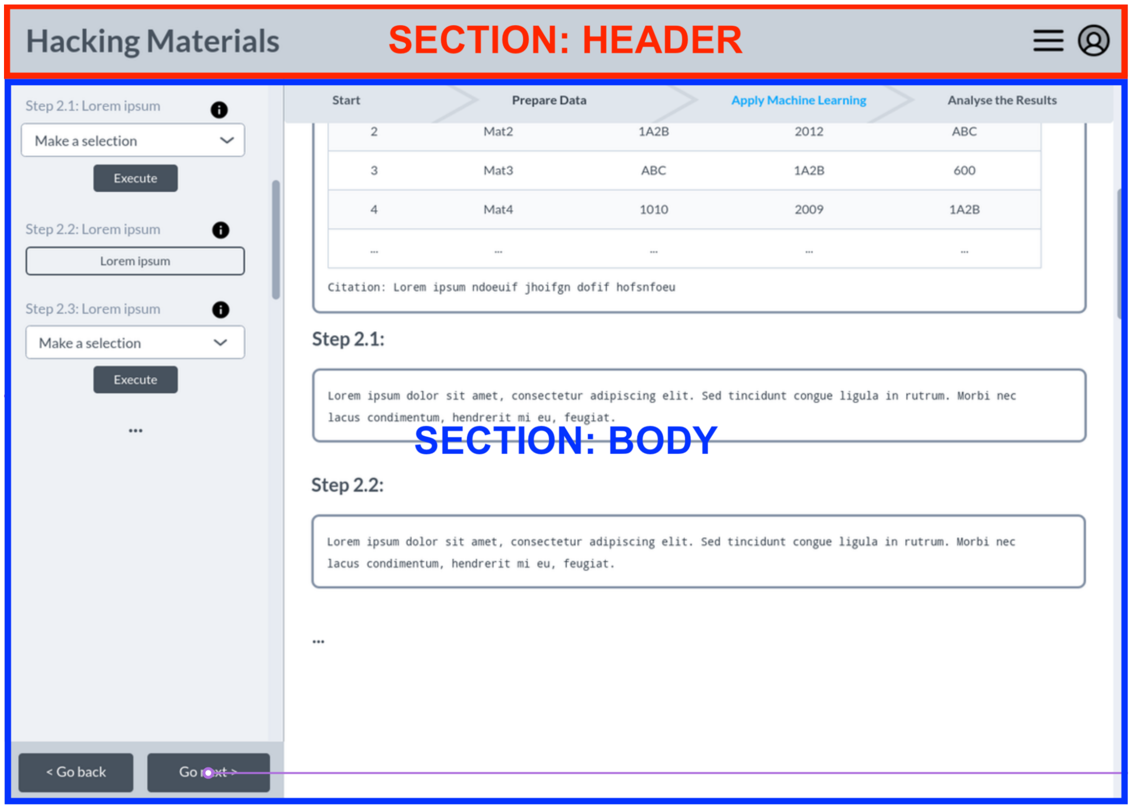
Sections

By referring to the low-fidelity prototype created earlier in the project, we divide the main application page into 2 main sections:

- Header: the top bar, which does not need to have context of what stage the user is up to and what's happening at any given point.
- Body: Includes 2 subsections that both need to know which stage the user is at (e.g. "Pre-process data" or "Apply machine learning"):
    - Left-side panel: named in the structure as `InputPanel`. Example code for this panel and how it shows the workflow steps is included in the sample code section below.
    - Main window on the right: named in the structure as `ViewingWindow`

AMENDMENT - 2 SEPTEMBER 2 2022

Felipe Lin (HA-BoxJelly) suggests potentially adding an `appFooter` as well, which allows visitors to find important information (like copyright) easily. No objections to this so far.

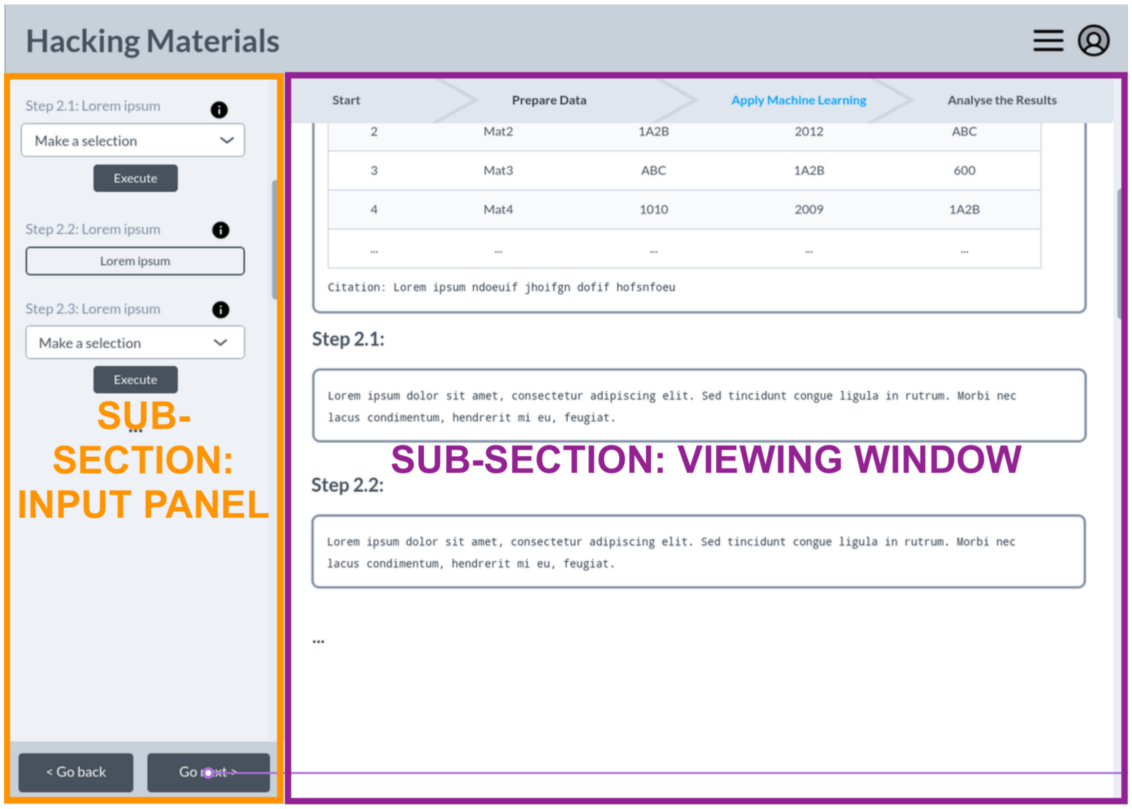


2	Mat2	1A2B	2012	ABC
3	Mat3	ABC	1A2B	600
4	Mat4	1010	2009	1A2B
...	...	...	...	...

Citation: Lorem ipsum ndoeuif jhoifgn dofif hofsnfoeu

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed tincidunt congue ligula in rutrum. Morbi nec lacus condimentum, hendrerit mi eu, feugiat.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed tincidunt congue ligula in rutrum. Morbi nec lacus condimentum, hendrerit mi eu, feugiat.



2	Mat2	1A2B	2012	ABC
3	Mat3	ABC	1A2B	600
4	Mat4	1010	2009	1A2B
...	...	...	...	...

Citation: Lorem ipsum ndoeuif jhoifgn dofif hofsnfoeu

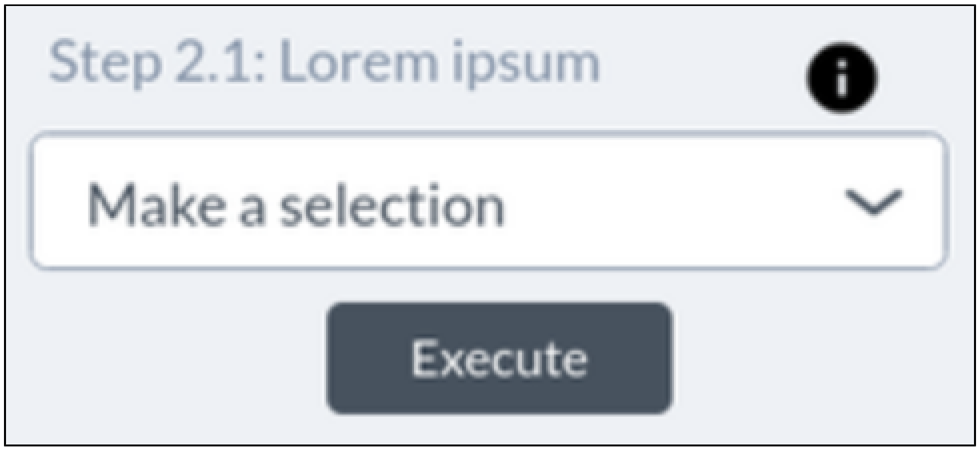
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed tincidunt congue ligula in rutrum. Morbi nec lacus condimentum, hendrerit mi eu, feugiat.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed tincidunt congue ligula in rutrum. Morbi nec lacus condimentum, hendrerit mi eu, feugiat.

Components

This folder will mainly contain all reusable components, e.g. Button, Tooltip, Modal, etc.

Notably, some of these reusable components would be "step types", e.g. `DropdownSelectStepType`. This example step type refers to the entire object shown below, including a step number, title, tooltip, dropdown list, button and whatever else may be needed. We would create this as a reusable component because many steps have similar requirements, e.g. selecting a dataset and selecting a featurizer should both be dropdown list type steps.



Steps

The word "steps" in this section refers specifically to the workflow steps that would be shown in the input panel, e.g. Dataset Selection step, Featurizer Selection step, etc.

A separate folder is created for these so that there would be a clear pattern that is easy to follow whenever more steps need to be added. Each step would use a step type component that is imported from the `/components` folder. E.g. the `DatasetSelectionStep` would use the `DropdownSelectStepTyp`
, as shown in the sample code snippet below.

Sample code snippets

Sample src/steps/datasetSelection/index.tsx

```
import DropdownSelectStepType from '../../components/dropdownSelectStepType';
import HelpModal from './HelpModal';
...
const DatasetSelectionStep = (props) => {
    ...
    const STEP_KEY = "dataset_selection"

    const options = api_call_here() // calls backend API to get the dataset options

    const onSubmit = selected_value => send_to_backend() // send to backend using api

    return (
        <DropdownSelectStepType
            stepNumber={props.stepNumber}
            title="Select Dataset"
            description="bla bla"
            tooltipContent={HelpModal}
            options={options}
            onSubmit={onSubmit}
        />
    );
};
```

Sample src/sections/appBody/InputPanel/index.tsx

```
import DatasetSelectionStep from '../../../steps/datasetSelectionStep';
import FeatuirzerSelectionStep from '../../../steps/featuirzerSelectionStep';
...

const InputPanel = (props) => {
    ...
    const { stage } = props;

    if (stage === 1) {
        return (
            <div>
                <DatasetSelectionStep
                    stepNumber="1.1"
                    data={data}
                    handleChange={handleChange}
                />
                <FeatuirzerSelectionStep
                    stepNumber="1.1"
                    data={data}
                    handleChange={handleChange}
                />
                ...
            </div>
        );
    } else if (stage === 2) {
        return (
            <div>
                ...
            </div>
        );
    }
};
```

| Deployment Process

Prerequisites:

a. Google API key for SSO login - Authenticate using API keys | Authentication | Google Cloud

b. Microsoft API key for SSO login - Enable authentication in a web API by using Azure Active Directory B2C | Microsoft Learn

c. Email Address of assigned database administrator

Worth a read:

- Connect over SSH with Visual Studio Code

- Docker - Visual Studio Marketplace

There are three steps to deploying this product (see videos below)

1. Spin up a VM on Nectar ARDC, Google Cloud, AWS or Azure. Ensure you choose an **Ubuntu 22.04 image.**

   Your browser does not support the HTML5 video element

1. Clone the GitHub repository and run the deployment script using the commands below.

```
sudo apt update && sudo apt upgrade && sudo apt install git
git clone https://github.com/COMP90082-2022-SM2/HA-2022-SM2.git
cd HA-2022-SM2/src/backend/
git checkout deployment
sudo chmod +x ./deploy_on_ubuntu22_04.sh
./deploy_on_ubuntu22_04.sh
```

   Your browser does not support the HTML5 video element

1. Point a domain name to the IP address of the VM (Unimelb IT department).

| Set-up SSO Authorisation Guide

hacking_material...thentication.pdf

| Specifications

Links
| Project Description
| Motivational Model
| Personas
| User Stories
| Prototype
| Business Case
| Plan
| Test cases
| Acceptance Criteria

| Project Description

Goals

- Help Materials engineers and people with interest in the field to predict the properties of new material by providing a tool that removes the complexity of performing Materials data retrievals and applying machine learning modelling.

Sponsor

- Dr Brandl
    - Lecturer in UOM. Teaches computational material engineering.
    - Has some knowledge of python and Jupyter Notebook, but knows very little about programming and algorithms.

Scope - User Types

- Regular user
    - General user with knowledge and interest in materials engineering.
    - Has no or very little experience in python, programming or machine learning modelling.
    - Would benefit from an easy-to-use application to assist in retrieving data and performing applied machine learning functionalities.

- Pro user
    - Regular user with access to additional features:
        - In control of the codes of the ML methods and the datasets.
            - Able to modify the python codes in the interface.
            - Able to introduce new datasets from new sources.
            - Able to add new features and delete existing features (the latter is optional).
        - Account management (not mentioned in the meeting but should have this function).
        - For now, the pro user is Dr Brandl himself.

Scope - Functionality

- Web page
    - Able to demonstrate diagrams.
    - Able to show more details about data entries in the diagram on click.
    - Provide hints and guidance for new users (low priority).
    - Provide an interface for pro users to code directly.
    - Quick authentication and sign-in for users that uses unimelb.edu.au domain emails.
- Backend server
    - At least allow 30+ users to operate concurrently for the first version.
    - Runs on Unimelb cloud server (i.e. Nectar).
- Machine learning module
    - Feature engineering
        - Able to generate a report about the importance of every feature in a specific dataset.
        - Analyze the relationship between different features (optional).
        - Allow pro users to add new features.
        - Allow all users to define which features to use.
    - Algorithms
        - Provide various ML algorithm options for users to choose from.
        - The logic can be exported into a Jupyter notebook file (the nature of the exported file may change in the future).
        - Pro users should be able to modify the ML algorithms and even upload their own scripts.

Technologies

- Hosting
    - University of Melbourne Cloud Services
- Frontend
    - React
    - Typescript
- Backend
    - Python
    - Flask
    - Docker
- Admin & Collaboration Tools
    - Confluence
    - Trello
    - GitHub
    - Slack

Stakeholders

Stakeholder	Role	Interests and expectations project	Project Influence / Importance
Dr. Christian Brandl	Main Customer	Receive a properly working application with at least the scope must-have features	High
Students from teams BoxJelly, Bluering, Redback	Product Developers	Acquire knowledge and experience. Fulfil subject's goals and hurdles	High
Subject Supervisor	Project Supervisor	Guide students to follow the good practices of software project development	Medium
Future application users	Client	Have access to a tool that facilitates data retrieval and application of Machine Learning in the field of Materials Science	Medium

Team Members

Name	Role
Dara O hEidhin	Backend Developer
Felipe Leefu Huang Lin	Frontend Developer
Radhimas Djan	Backend Developer
Yaoming Xuan	Machine Learning Developer
Zhaoqi Wang	Team Representative & Developer

Additional Details

- Dataset is clean and reliable. It shouldn't require much preprocessing.
- The project should utilise the Matminer material analysis library.

Change log

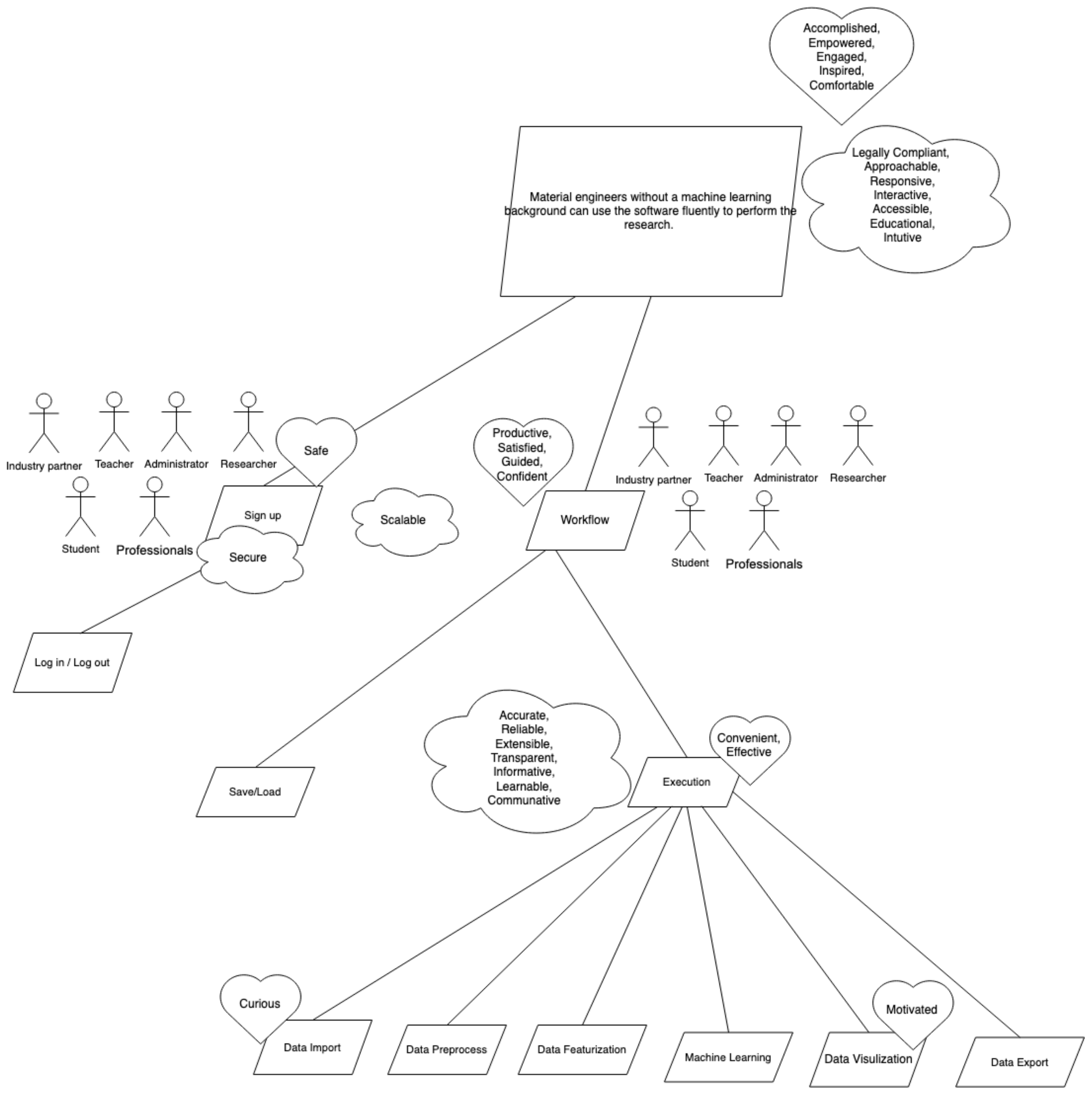
Date	Version	Author	Comment
15 Aug 2022	1.0	Felipe Leefu Huang Lin	First draft
19 Sep 2022	1.1	Dara O hEidhin	Changes based on Mauro's feedback
19 Oct 2022	1.2	Felipe Leefu Huang Lin	Changes based on Mauro's sprint 2 feedback
20 Oct 2022	1.3	Felipe Leefu Huang Lin	Added team members' information

| Motivational Model

Do-Be-Feel List - Cross-Teams - Version 2.0

Who (users)	Do (functional goals)	Be (qualitative goals)	Feel (emotional goals)
Students	Add more databases, machine learning methods and plot types	Accessible	Accomplished
Administrators	Compare data using tables & plots	Accurate	Comfortable
Professionals	Data Pre-processing: Calculate descriptive statistics	Approachable	Confident
Industry Partners	Data Pre-processing: Consider anonymized data	Communicative	Convenient
Teachers	Data Pre-processing: Overview of the current import data	Educational	Curious
Researcher	Data Pre-processing: Reduces noise and eliminates ambiguity	Extensible	Effective
Code maintainers	Data Pre-processing: Standardizing data to bring it into the formatting range	Informative	Empowered
	Data Visualization: Data processing: Tabular data & Plotted Graph	Interactive	Engaged
	Edit python code directly in the interface	Intuitive	Guided
	Export input data	Learnable	Inspired
	Export Jupiter notebook file	Legally Compliant	Motivated
	Export output data tables and figures	Reliable	Productive
	Featurization data: Add multiple composition-based features	Responsive	Safe
	Featurization data: Add multiple simple density features	Scalable	Satisfied
	Import Data: Create working spaces when importing	Secure	
	Import Data: Drag and drop the import of files	Transparent (progress, error messages, notebook export...)	
	Import Data: Import data files (CSV, XES, Parquet) from the local system		
	Log in/Log out		
	Machine Learning: Define input data and output data: Splitting data into training, test, and validation sets		
	Machine Learning: Determining model features and training the model: Configure and adjust hyperparameters for optimum performance		
	Machine Learning: Evaluate model performance and establish benchmarks: Continuous measurement and monitoring of model performance		
	Machine Learning: Evaluate model performance and establish benchmarks: Evaluate models using validation methods and validation datasets		
	Machine Learning: Get model results: The most important features of the current ML model		
	Machine Learning: Select the machine learning model to be used		
	Maintain software		
	save/load workflows		
	Sign up		

Goal Model - Cross-Teams - Version 2.0


Accomplished, Empowered, Engaged, Inspired, Comfortable
Material engineers without a machine learning background can use the software fluently to perform the research.
Legally Compliant, Approachable, Responsive, Interactive, Accessible, Educational, Intutive
Industry partner
Teacher
Administrator
Researcher
Student
Professionals
Safe
Sign up
Secure
Scalable
Log in / Log out
Productive, Satisfied, Guided, Confident
Workflow
Save/Load
Accurate, Reliable, Extensible, Transparent, Informative, Learnable, Communative
Convenient, Effective
Execution
Curious
Data Import
Data Preprocess
Data Featurization
Machine Learning
Motivated
Data Visulization
Data Export

Change log

Version date	Editor	Comment
08 Aug 2022	Dara O hEidhin	Initial template (version 1.0)
15 Aug 2022	Yaoming Xuan	Fill in the information about the students (version 1.0)
17 Aug 2022	Dara O hEidhin	Merged do-be-feel and goal model into one page (version 1.0)
17 Aug 2022	Felipe Leefu Huang Lin	Restructure the Do-Be-Feel list (version 1.0)
17 Aug 2022	Felipe Leefu Huang Lin	Upload the first draft of the goal model (version 1.0)
18 Aug 2022	Zhaoqi Wang	Filled in some blanks (version 1.0)

18 Aug 2022	Zhaoqi Wang	Added row on Do-Be-Feel list (version 1.0)
18 Aug 2022	Dara O hEidhin	Added row on Do-Be-Feel list (version 1.0)
18 Aug 2022	Radhimas Djan	Added row on Do-Be-Feel list (version 1.0)
21 Aug 2022	Zhaoqi Wang	Updated with merged results (Final) with other teams (version 2.0)
22 Aug 2022	Felipe Leefu Huang Lin	Added 3 teams' collaboration Do-Be-Feel list and Goal Model (version 2.0)
19 Oct 2022	Felipe Leefu Huang Lin	Changes according to Mauro's sprint 2 feedback. (Removed multiple versions of content)

| Personas

Persona 1 (Student User) - Prepared by team BoxJelly

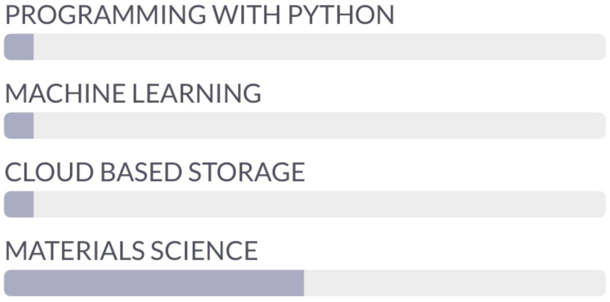
Assol Anahita

age:	22
residence:	Melbourne
education:	Materials Science and Engineering
occupation:	Student
marital status:	Single

"It's SO time consuming to do material research and get decent results through just a semester."

Motivation: As a materials engineering graduate student, Assol gets frustrated and demotivated when she can't make sense of the data she has and feels that she is not really making real progress with her studies. She needs a tool that can speed up materials data retrieval and processing so she can focus in analysing the results to better understand the theory and concepts of materials engineering.

Comfort With Technology



Criteria For Success:

Assol can perform materials data requests/retrievals and accurate materials property predictions supported by Machine Learning technology with easy to follow steps button clicks user interface.

Needs

- Easy-to-use interface materials science data processing and retrieval application
- A tool to predict property of a material with assistance of Machine Learning technology without prior knowledge of Python and Machine Learning programming

Wants

- A data mining application that helps her research projects
- A better understanding on how Machine Learning can help her to learn more about a material
- Ability to use ML algorithms as a black box
- Freedom to select features on her own terms
- A tool to accelerate research progress

Values

- Convenience
- Quickness
- Safety
- Understandable

Fears

- Spends hours working on a research project with very little progress because she neither has an adequate tool to do data mining, nor the programming skill to analyse the data herself
- Have to conduct countless experiments to figure out the properties of the materials
- Hard to choose suitable ML algorithms

Persona 2 (Regular User) - Prepared by team BlueRing

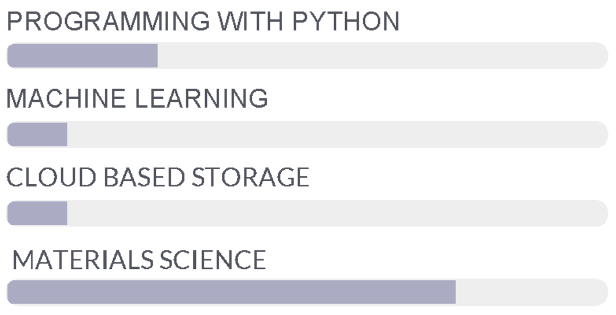
Gray Zhou

age:	28
residence:	Ningde, Fujian, China
education:	Master of Material Engineering
occupation:	R & D Engineer of Polymer
marital status:	Single

"It is fantastic to apply a multi-function online tool with ML methods if it is efficient and reliable. Nobody will refuse a tool that can save time"

Motivation: Gray Zhou is a R & D Engineer of polymers in a battery factory. His work is searching for better materials for battery production. Gray spends a lot of time testing different materials, but some of tests are waste of time because of the poor performance observed. He needs a system that can predict some useful properties of materials so that he can remove samples with low predicted performance and boost the research. His company provides some ML tools, but they are awkward and only have limited functions.

Comfort With Technology



Criteria For Success:

Provide a website or online-tool with quick, visual interface which can help him in daily development of new materials.

A successful product should help him save noticeable time on data processing and provide reliable prediction of properties.

Needs

- Retrieve and extract required data, process the data with ML methods to get some properties
- Provide graphs which can be modified with interface about predicted properties
- Help finding the material with best predicted properties

Wants

- Ability to interact with the graph to further compare serval materials in detail
- Upload data from his lab for predicting
- Explain what ML method the system applied and how it helps the prediction
- Continue his work on mobile devices without gaps of interaction

Values

- Easy to get started on both desktop and mobile
- Efficient back-end process
- Abilities to select functions and filter results
- Well organized visualization of interface and graphs

Fears

- Not enough guidance in the web or tool so him may feel confused to find functions he wants.
- Lacking understand of what the system done, then reducing the confidence level of his report
- Frequently unable to access the system

Persona 3 (Pro User) - Prepared by team RedBack

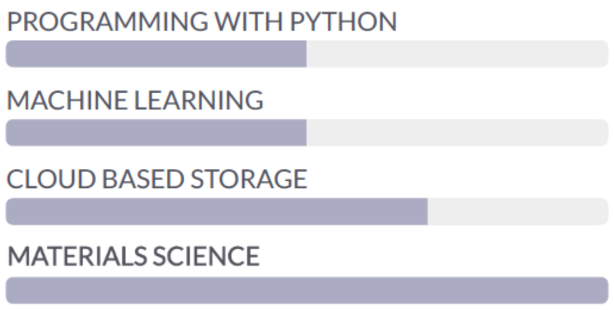
Alex

age:	45
residence:	Melbourne
education:	Master's Degree in Physics
occupation:	Materials Engineer
marital status:	Divorced without kids

"There has got to be a better way to do this."

Motivation : As an experienced Materials Engineer, Alex's job requires him to narrow down candidate materials by performing physical experiments to choose a material which can takes years to do. He needs a tool that can speed up the process by narrowing down candidate materials for experimentation using Machine Learning and simulations.

Comfort With Technology



Criteria For Success:

Alex can find the right materials efficiently, with accurate results and that matches the client's requirements.

Needs

- Products to accelerate his workflow
- Access to wide variety of related tools and resources

Wants

- Suitable models and featurizers for different use cases
- Demonstrate reproducible results to his clients
- Share resources with others
- Refining generated workflow to reuse

Values

- Extensibility
- Accuracy
- Reliability
- Responsiveness
- Scalability
- Transparency

Fears

- Tool is too inflexible
- Losing access to progress on his work
- Not being able to verify his results
- Not having support with the tool

Change log

Date	Version	Author	Comment	File
15 Aug 2022	1.1	Felipe Leefu Huang Lin	First draft	persona.pdf
16 Aug 2022	1.2	Yaoming Xuan	version 2 of the persona	persona.pdf
16 Aug 2022	1.3	Felipe Leefu Huang Lin	Used name generator to create a random name	student.pdf
16 Aug 2022	1.4	Yaoming Xuan	Converted to docx file	persona....960.docx

16 Aug 2022	1.5	@Yaoming Xuan	small modifications	persona ...r15.docx
17 Aug 2022	1.6	Zhaoqi Wang	Suggestions on Quote, Needs, Wants, Values, Fears and Logo (Brand). Issues: 1. can't upload more picture 2. extended to 2 pages	Assol Ana...rator.pdf
20 Aug 2022	1.7	Yaoming Xuan	Generate a pdf file with latest content	persona ver17.pdf
22 Aug 2022	1.8	Zhaoqi Wang	small modifications	
22 Aug 2022		Felipe Leefu Huang Lin	Added personas created by teams BlueRing and RedBack	
22 Aug 2022		Dara O hEidhin	Embedded pngs of our personas for site PDF export compatibility	
18 Sep 2022	1.9	Felipe Leefu Huang Lin	Revised persona following sprint 1 feedback.	persona_1.9.pdf
18 Sep 2022	2.0	Felipe Leefu Huang Lin	Updated newest version of team personas	
20 Oct 2022	3.0	Felipe Leefu Huang Lin	Updated student and pro user persona	student_...sona.pdf

Link to make additional changes: https://personagenerator.com/7300af72-1c96-11ed-8d77-d742c7efee51/7300af73-1c96-11ed-8d77-5f5915d1141f

| User Stories

Prioritization Technique

We used the MoSCoW prioritization classification.
Must have - must be included in the scope of the project, we defined this all the must-have user stories can create a minimum viable product
Should Have - should be included in the scope of the project
Could Have - could be included in the scope of the project
Won't Have - will not be included in the scope of the project

Epics & Owning team allocation

	Epic	Total Size	Highest Priority within Epic	Assigned Team
2	**Administration**	31	1 - Must Have	BoxJelly
4	**Data Visualisation**	13	1 - Must Have	BlueRing
6	**External Data**	8	2 - Should Have	Unassigned - stretch goal
1	**Input Data**	27	1 - Must Have	RedBack
5	**Jupyter Notebook**	20	2 - Should Have	Unassigned - stretch goal
3	**Machine Learning**	16	1 - Must Have	BlueRing

Teams Collaboration - Version 3.0

ID		Role		Action	Epic		Goal	Size (days)	Priority	Assigned Team
30	As a	general user	I want to	be able to view the citations for used featurizers	Input Data	so that	I could know more about the source of the featurizers (legally compliant)	1	1 - Must have	RedBack
32	As a	general user	I want to	browse and select built-in featurizers	Input Data	so that	I can discover ways of manipulating my data	1	1 - Must have	RedBack
34	As a	general user	I want to	browse built-in datasets	Input Data	so that	I can discover data to experiment with	1	1 - Must have	RedBack
19	As a	student	I want to	quickly browse the Materials available in the database for retrieval and simulations	Input Data	so that	I can quickly perform queries.	3	2 - Should have	RedBack
21	As a	general user	I want to	be able to select datasets from existing databases	Input Data	so that	I do not have to worry about how the data is loaded	3	1 - Must have	RedBack
37	As a	general user	I want to	be able to preview the output of each execution	Input Data	so that	I could explore the data	1	1 - Should have	RedBack
41	As a	general user	I want to	view the columns of the saved dataset and select the column I want to featurize	Input Data	so that	I can provide the correct input to the featurizer	3	1 - Must have	RedBack
25	As a	general user	I want to	Select specific features from a dataset	Input Data	so that	I can improve the precision of my model	3	2 - Should have	RedBack
13	As a	Pro user	I want to	add new features	Input Data	so that	they can be reused in the future	5	2 - Should have	RedBack
28	As a	general user	I want to	be able to reference/view citations for original data sources	Input Data	so that	I can retrieve data.	1	3 - Could have	RedBack
18	As a	pro user	I want to	be able to apply new featurizers	Input Data	so that	I can create new features	3	3 - Could have	RedBack
1	As a	student	I want to	clean and tune data input	Input Data	so that	I have less noise on visualizations.	5	3 - Could have	RedBack
29	As a	student	I want to	save project-specific data/checkpoints	Administration	so that	I can pick up where I left off for specific projects	1	1 - Must have	BoxJelly

35	As a	pro user	I want to	export model selections, parameters, and data flows	Administration	so that	I can save my work and share it with others	1	1 - Must have	BoxJelly
36	As a	pro user	I want to	import exported model selections, parameters, and data flows	Administration	so that	I can continue the work I had previously saved	1	1 - Must have	BoxJelly
20	As a	student	I want to	Create an account using single-sign-on, restricted to the *.unimelb.edu.au domain	Administration	so that	my research remains secure	3	1 - Must have	BoxJelly
23	As a	pro user	I want to	Control job execution	Administration	so that	I can **start, view the progress of**, and cancel jobs related to my project	3	1 - Must have	BoxJelly
10	As a	pro user	I want to	be able to use pro-user features	Administration	so that	I can access pro-user features	5	1 - Must have	BoxJelly
38	As a	pro user	I want to	have my pro user settings persist on each visit	Administration	so that	I don't have to reconfigure settings to use the features I need	1	2 - Should have	BoxJelly
24	As a	student	I want to	receive provided hints and guidance for new users	Administration	so that	I can quickly learn how to use software	3	2 - Should have	BoxJelly
14	As a	pro user	I want to	easily find and read the documentation on the pro features	Administration	so that	I can use them with ease	5	2 - Should have	BoxJelly
17	As a	pro user	I want to	Be kept informed about job status	Administration	so that	I can avoid polling my workspace to check for results	3	3 - Could have	BoxJelly
6	As a	pro user	I want to	have access to more processing power	Administration	so that	I can run more complex operations or use more data	5	3 - Could have	BoxJelly
31	As a	general user	I want to	able to select a Machine Learning model	Machine Learning	so that	I could use it to train and run the data	1	1 - Must have	BlueRing
33	As a	general user	I want to	browse built-in ML models	Machine Learning	so that	I can discover ways of manipulating my data	1	1 - Must have	BlueRing
39	As a	user	I want to	be able to select the split ratio of data	Machine Learning	so that	to train and test the model	1	2 - Should have	BlueRing
26	As a	pro user	I want to	have the option to change the hyperparameters used in the Machine Learning model	Machine Learning	so that	I can fine-tune my test results.	3	2 - Should have	BlueRing
15	As a	pro user	I want to	be able to use additional ML models	Machine Learning	so that	I can improve the accuracy	5	2 - Should have	BlueRing
7	As a	pro user	I want to	combine multiple ML models together	Machine Learning	so that	I can model more complex data manipulations	5	3 - Could have	BlueRing
40	As a	general user	I want to	select features used for x and y axis	Data Visualisation		I can plot the chart based on features I'm interested in	1	1 - Must have	BlueRing
22	As a	general user	I want to	see clear annotation or explanation of data points and features	Data Visualisation	so that	I can understand the results of the analysis	3	1 - Must have	BlueRing
8	As a	student	I want to	use different types of plotting graphs	Data Visualisation	so that	I have the flexibility to visualize data according to my needs.	5	1 - Must have	BlueRing
9	As a	general user	I want to	able to view and plot the results of the model	Data Visualisation	so that	I could analyze and visualise the effects of the model	5	1 - Must have	BlueRing
12	As a	student	I want to	export my work to a Jupyter Notebook	Jupyter Notebook	so that	I can extend my work beyond the capability of the application	5	2 - Should have	TBD - after the completion of all assigned "should have"s
2	As a	general user	I want to	attach comments to workflow objects	Jupyter Notebook	so that	I can document my work	5	3 - Could have	TBD - after the completion of all assigned "could have"s
4	As a	Pro user	I want to	edit python code on the interface	Jupyter Notebook	so that	I can control how the ML algorithms works	5	3 - Could have	TBD - after the completion of all assigned "could have"s
5	As a	Pro user	I want to	upload my own script (in python) if possible	Jupyter Notebook	so that	I can extend the tool to support custom models and featurizers	5	3 - Could have	TBD - after the completion of all assigned "could have"s
27	As a	pro user	I want to	be able to access new databases	External Data	so that	I can access additional data	3	2 - Should have	TBD - after the completion of all assigned "should have"s

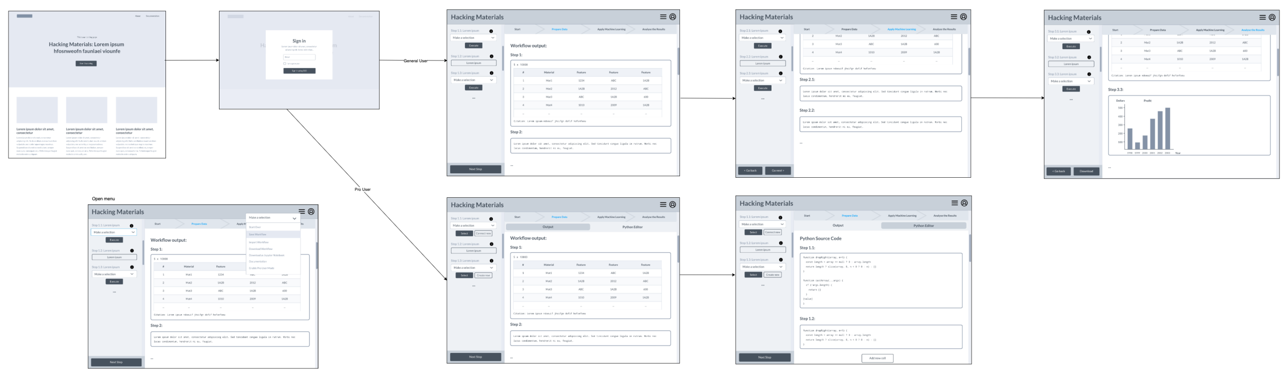
3	As a	Pro user	I want to	be able to add new datasets in the future	External Data	so that	if there's a new dataset that can be used on a new project, it can be added instantly	5	3 - Could have	TBD - after the completion of all assigned "could have"s
11	As a	student	I want to	analyze the relationship between different features		so that	I can identify which features I need to select for my analysis	5	2 - Should have	TBD - after the completion of all assigned "should have"s
16	As a	general user	I want to	add specific materials to the workflow for the analysis		so that	compare the performance of the specific material my client or I choose with other material	3	3 - Could have	TBD - after the completion of all assigned "could have"s

Change log

Version date	Editor	Comment
08 Aug 2022	Dara O hEidhin	Initial template
17 Aug 2022	Felipe Leefu Huang Lin	Added the first version of user stories
18 Aug 2022	Dara O hEidhin	Entered user stories in a table and added some more
18 Aug 2022	Zhaoqi Wang	Added more user stories
18 Aug 2022	Yaoming Xuan	Added pro-user stories
18 Aug 2022	Radhimas Djan	Added more user stories and edited some of the user stories
19 Aug 2022	Radhimas Djan	Added merged user stories
21 Aug 2022	Zhaoqi Wang	Version 3 user stories
22 Aug 2022	Felipe Leefu Huang Lin	Added the final version of collaboration user stories.
22 Aug 2022	Zhaoqi Wang	Added "assigned team" column
18 Sep 2022	Zhaoqi Wang	Implemented changes according to the feedback
18 Sep 2022	Dara O hEidhin	Deleted old versions of the user stories table and only kept the current table and changelogs
23 Sep 2022	Zhaoqi Wang	Updated user stories
04 Oct 2022	Zhaoqi Wang	Added user story 41
19 Oct 2022	Felipe Leefu Huang Lin	Update user stories according to Mauro's sprint 2 feedback

| Prototype

Teams Collaboration - version 1.0 (Final)



Descriptive Notes (recorded by team redback)

- Landing page:
    - Static page with information about Matminer and machine learning. This page may also have a link to the user manual in the future.
    - Link to access the app
        - On click, it opens a login modal
        - Once the users are logged in, they're redirected to the app
- Single page app:
    - Top bar:
        - The "User profile" button at the top opens a menu to give the user the option to log out
        - The "Menu" button at the top has options to import or save a workflow, download it in different formats, start over, a link to the documentation and a toggle to enable the pro view.
    - General user:
        - The workflow is divided into major and minor steps. Each major step would have its own page. Users can go back and forth between the major steps as needed.
        - Left panel:
            - All the minor steps are numbers and named to guide the user
            - Inputs can be of different types
            - Each step has a tooltip button that opens a modal with guidance information about the step
            - The steps and options in the left panel should always be the same, no matter what selections the user made in previous steps. Any step that requires customized inputs would open in a window..
                - Example 1: Step 3.1 might be "Selecting a plot type". As there is a known, limited list of different plot types, this step may be a drop-down menu that is displayed directly in the left panel.
                - Example 2: Step 3.2 might be customizing the selected plot's configuration options. As different plot types may need different configuration options, these options will not be displayed in the panel directly. Instead, the panel will include only a button that says "Configure plot", which would open a modal with the specific options applicable to the selected plot type.
            - Pinned buttons at the bottom of the panel: navigate between the different major steps. The last step page may also have a button to download the full workflow.
        - Viewing window:
            - At the top of the viewing window, the user can see the progression of major steps with the current step highlighted.
            - The output of each minor step is labelled with the step number and contained inside a box. The output inside the box is the same output produced by running the python code, simply copied over for transparency.
            - The outputs from the previous pages are also always displayed, so it's not just the outputs of the current page.
            - Where a resource with citations is used, the citations will be automatically printed after the output of the step where the resource was selected.
    - Pro user:
        - Left panel: has all the same options as a general user, plus additional buttons to configure their own settings as needed
        - Viewing window: the window has 2 tabs:
            - Output: same the as the viewing window of the general user
            - Python source code:
                - An editable view of all the code generated by their selections, which looks similar to a Jupyter notebook.
                - Users can add new cells as desired
                - Brings up the following question: what happens if the user edits the code generated by one of the steps? This may lead to inconsistencies between what is shown in the step's input field and what the code now actually does. This is an implementation decision so is not a major concern right now, but one option that we decided to show in the prototype is that the step's input in the left panel would change to say "Custom" or something similar, indicating that the configuration was changed.

User Testing Prototypes

After we divided the workflow steps into three stages, it became unclear when and how the user would execute the workflow. After asking the client what he preferred, we realised that this is a more complex decision than we expected and it would need more consideration. To help the client reach a decision, we prepared three functional low-fi prototypes for user testing. Each prototype represented a different approach. The approaches were as follows:

- Approach A - Run workflow steps individually. Prototype: https://marvelapp.com/prototype/2g445gdg/screen/88805093
- Approach B - Run the full workflow at once at the end. Prototype: https://marvelapp.com/prototype/2g445gdg/screen/88805697
- Approach C - Run each stage (which includes a number of steps) individually. Prototype: https://marvelapp.com/prototype/2g445gdg/screen/88805736

The prototypes were shared with the client, but as he was unavailable at the time, we decided to preemptively adopt Approach C as it offered the best balance between them. Once the client was able to respond, he agreed with our decision. We then implemented Approach C, using the prototype as a guide.

Change log

Version date	Editor	Comment
20 Aug 2022	Felipe Leefu Huang Lin	First prototype draft was made in collaboration with team Redback and Bluering
21 Oct 2022	Zhaoqi Wang	Added user-testing prototypes

| Plan

Options:

Scenario 1 - If the projected is divided between the three teams then we expect to split each epic equally between the three teams.

Scenario 2 - If we work as separate teams the user stories will be addressed in order of priority and entered into Trello. Please refer to Business Case and User Stories

Collaborative changes, on GitHub, will pass a reviewing process with each sprint being released with a tag.

Result:

It was agreed to follow Scenario 1

Change Log

Date	Version	Author	Comment
15 Aug 2022	1.0	Redback Team	First draft
19 Sep 2022	1.1	Dara O hEidhin	Changes based on Mauro's feedback

| Test cases

US10, TC01: Login with the pro user(successful)

Test Type: Functional	**Execution type:** Manual
Objective: Pro users are able to log in with a pro-user feature	
Setup The databases have been created, and a google API key has been registered for the project	
Pre-condition 1. A registered user has a pro-user access	
Notes: [1]: Click on Start exploring on the home page *application pop-up a google SSO [2]: When a google pop-up reveals, tick the "i am pro user". *application shows ticked marked on "i am pro user" [3]: Click sign in with google	
Time constraint: Minimum: 10 seconds Maximum: 1 minute	

US 20, TC01: create a new account (successful)

Test Type: Functional	**Execution type:** Manual
Objective: Users are able to create an account using an SSO with his/her student account.	
Setup The databases have been created, and a google API key has been registered for the project	
Pre-condition 1. The user does not have an account on the databases.	
Notes: [1]: Click on Start exploring on the home page *application pop-up a google SSO [2]: When a google pop-up reveals, click on login using google, then it will create an account using the student email account. *application redirects to the /workflow	
Time constraint: Minimum: 10 seconds Maximum: 1 minute	

US 20, TC02: existing user login (successful)

Test Type: Functional	**Execution type:** Manual
Objective: Users are able to log on to an account using SSO with his/her student account.	
Setup The databases have been created, and a google API key has been connected to the system	
Pre-condition 1. The user has an account on the databases.	
Notes [1]: Click on Start exploring on the home page *application pop-up a google SSO [2]: When a google pop-up reveals, click on login using google, then log in using the student email account. *application redirects to the /workflow	
Time constraint: Minimum: 10 seconds Maximum: 1 minute	

US 29, TC01: Save selected dataset (successful)

Test Type: Functional	**Execution type:** Manual
Objective: The dataset user selected is saved successfully.	
Setup The database has been established in the back end.	
Pre-condition 1. The user has logged in. 2. The user selected a dataset at step 1.1	
Notes: [1]: Click on the save button at step 1.1 * Application shows “Step 1.1 output:” * Application shows a green box where a notice saying, “Your selection was successfully saved”	
Time constraint: Minimum: 1 second Maximum: 10 seconds	

US 29, TC02: Save selected dataset column (successful)

Test Type: Functional	**Execution type:** Manual
Objective: The dataset column user selected is saved successfully.	
Setup The database has been established in the back end.	
Pre-condition 1. The user saved the dataset successfully. 2. The user selected a dataset column at step 1.2	
Notes [1]: Click on the save button at step 1.2 * Application shows “Step 1.2 output:” * Application shows a green box where a notice saying, “Your selection was successfully saved”	
Time constraint: Minimum: 1 second Maximum: 10 seconds	

US 29, TC03: Save selected featurizer (successful)

Test Type: Functional	**Execution type:** Manual
Objective: The featurizer user selected is saved successfully.	
Setup The database has been established in the back end.	
Pre-condition 1. The user saved the dataset column successfully. 2. The user selected a featurizer at step 1.3	
Notes [1]: Click on the save button at step 1.3 * Application shows “Step 1.3 output:” * Application shows a green box where a notice saying, “Your selection was successfully saved”	
Time constraint: Minimum: 1 second Maximum: 10 seconds	

US 29, TC04: Save selected machine learning model (successful)

Test Type: Functional	**Execution type:** Manual

Objective: The machine learning model user selected is saved successfully.
Setup The database has been established in the back end.
Pre-condition 1. The user saved the featurizer successfully. 2. The user selected a machine learning model at step 2.1
Notes [1]: Click on the save button at step 2.1 * Application shows “Step 2.1 output:” * Application shows a green box where a notice saying, “Your selection was successfully saved”
Time constraint: Minimum: 1 second Maximum: 10 seconds

US 29, TC05: Save selected property (successful)

Test Type:	**Execution type:**
Functional	Manual
Objective: The property user selected is saved successfully.	
Setup The database has been established in the back end.	
Pre-condition 1. The user saved the machine learning model successfully. 2. The user selected property at step 2.2	
Notes [1]: Click on the save button at step 2.2 * Application shows “Step 2.2 output:” * Application shows a green box where a notice saying, “Your selection was successfully saved”	
Time constraint: Minimum: 1 second Maximum: 10 seconds	

US 29, TC06: Save selected x-axis feature (successful)

Test Type:	**Execution type:**
Functional	Manual
Objective: The x-axis feature user selected is saved successfully.	

Setup The database has been established in the back end.
Pre-condition 1. The user saved the property successfully. 2. The user selected an x-axis feature at step 3.1
Notes [1]: Click on the save button at step 3.1 * Application shows “Step 3.1 output:” * Application shows a green box where a notice saying, “Your selection was successfully saved”
Time constraint: Minimum: 1 second Maximum: 10 seconds

US 29, TC07: Save selected y-axis feature (successful)

Test Type: Functional	**Execution type:** Manual
Objective: The y-axis feature user selected is saved successfully.	
Setup The database has been established in the back end.	
Pre-condition 1. The user saved the x-axis feature successfully. 2. The user selected a y-axis feature at step 3.2	
Notes [1]: Click on the save button at step 3.2 * Application shows “Step 3.2 output:” * Application shows a green box where a notice saying, “Your selection was successfully saved”	
Time constraint: Minimum: 1 second Maximum: 10 seconds	

US 29, TC08: Save selected regression function (successful)

Test Type: Functional	**Execution type:** Manual
Objective: The regression function user selected is saved successfully.	
Setup The database has been established in the back end.	

Pre-condition 1. The user saved the y-axis feature successfully. 2. The user selected a regression function at step 3.4
Notes [1]: Click on the save button at step 3.4 * Application shows “Step 3.4 output:” * Application shows a green box where a notice saying, “Your selection was successfully saved”
Time constraint: Minimum: 1 second Maximum: 10 seconds

US35, TC01: Export workflow (Successful)

Test Type: Functional	**Execution type:** Manual
Objective: Exporting current workflow into .json state	
Setup The databases have been created, a google API key has been registered for the project, and the user has logged on.	
Pre-condition 1. A registered and logged-in user 2. The user has selected workflow step 1. Prepare Data and 2. Machine learning step that he want to save as a json file	
Notes: [1]: Click the export workflow *Application downloads the JSON step of the workflow into the download folder user desired	
Time constraint: Minimum: 10 seconds Maximum: 1 minute	

US36, TC01: Import workflow (successful)

Test Type: Functional	**Execution type:** Manual
Objective: Importing a .json file into the workflow	
Setup The databases have been created, a google API key has been registered for the project, and the user has logged on.	
Pre-condition 1. A registered and logged-in user	

Notes: [1]: Click import workflow *Application open a windows folder for user to upload their saved workflow in .json format [2]: Click the desired .json files *Application will have all the .json
Time constraint: Minimum: 10 seconds Maximum: 1 minute

| Acceptance Criteria

ACCEPTANCE CRITERIA

USER STORY ID	USER STORY	GIVEN	WHEN	THEN
20	Create User Database and Endpoints	I have a registered account for use of this product	I update any relevant personal information	That updated information persists in the database and is linked to my account
			I make changes to my workflow	That workflow data can be saved in the database and linked to my account. Only I can access it
			I delete my account	All linked data is deleted from the database. Others' data won't be affected
20	Create an account using single-sign-on, restricted to the *.unimelb.edu.au domain	I own a Unimelb account and would like to access the web application features	I click on the login button	I can sign in using my unimelb.edu.au email as a form of authentication
		I am a first-time user and don't own a Unimelb account	I click on the login button	If I try to sign in using a non-Unimelb email I receive a notification that only unimelb.edu.au emails are allowed
29	Save project specific data/checkpoints	I am logged into my account	I make changes to my workflow	That data can be saved to the database under my user account
35	Export model selections, parameters, and data flows	I am logged into my account	I click on the export button	I can download a file that has all the data needed to recreate my current workflow state
36	Import exported model selections, parameters, and data flows	I am logged into my account	I click on the import button	I can upload a previously exported workflow state
		I have uploaded a previously exported workflow state	The file passes all data integrity checks	My UI will match the state represented in the imported file
10	opt in to pro-user features	I am a pro user and would like to access my pro-user features	I click on the opt button	I jump into a new window with all pro-user features accessible.
23	Control job execution	I have a task in progress	I click on the start, view and cancel buttons	The running tasks can be started, viewed and cancelled

Change Log

Date	Version	Author	Comment
17 Sep 2022	1.0	Felipe Leefu Huang Lin	First draft
19 Sep 2022	1.1	Dara O hEidhin	Added user stories from Sprint 2

| Sprint Artefacts

Links
- Trello Board link
- Sprint Review
- Final Presentation
- Retrospectives
- Product planning
- Client Communications

Trello Board link

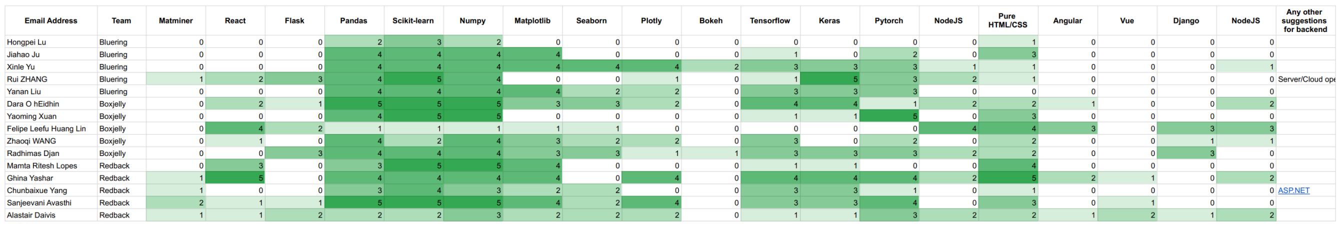
https://trello.com/b/PzZuNQMk/boxjelly-sprint

| Final Presentation

HA final presentation.pdf

Resources

Team HA - Skills assessment



Email Address	Team	Matminer	React	Flask	Pandas	Scikit-learn	Numpy	Matplotlib	Seaborn	Plotly	Bokeh	Tensorflow	Keras	Pytorch	NodeJS	Pure HTML/CSS	Angular	Vue	Django	NodeJS	Any other suggestions for backend
Hongpei Lu	Bluering	0	0	0	2	3	2	0	0	0	0	0	0	0	0	1	0	0	0	0	
Jiahao Ju	Bluering	0	0	0	4	4	4	4	0	0	0	1	0	2	0	3	0	0	0	0	
Xinle Yu	Bluering	0	0	0	4	4	4	4	4	4	2	3	3	3	1	1	0	0	0	1	
Rui ZHANG	Bluering	1	2	3	4	5	4	0	0	1	0	1	5	3	2	1	0	0	0	0	Server/Cloud ope
Yanan Liu	Bluering	0	0	0	4	4	4	4	2	2	0	3	3	3	0	0	0	0	0	0	
Dara O hEidhin	Boxjelly	0	2	1	5	5	5	3	3	2	0	4	4	1	2	2	1	0	0	2	
Yaoming Xuan	Boxjelly	0	0	0	4	5	5	0	0	0	0	1	1	5	0	3	0	0	0	0	
Felipe Leefu Huang Lin	Boxjelly	0	4	2	1	1	1	1	1	0	0	0	0	0	4	4	3	0	3	3	
Zhaoqi WANG	Boxjelly	0	1	0	4	2	4	3	2	2	0	3	0	2	0	2	0	0	1	1	
Radhimas Djan	Boxjelly	0	0	3	4	4	4	3	3	1	1	3	3	3	2	2	0	0	3	0	
Mamta Ritesh Lopes	Redback	0	3	0	3	5	5	4	0	0	0	1	1	0	0	4	0	0	0	0	
Ghina Yashar	Redback	1	5	0	4	4	4	4	0	4	0	4	4	4	2	5	2	1	0	2	
Chunbaixue Yang	Redback	1	0	0	3	4	3	2	2	0	0	3	3	1	0	1	0	0	0	0	ASP.NET
Sanjeevani Avasthi	Redback	2	1	1	5	5	5	4	2	4	0	3	3	4	0	3	0	1	0	0	
Alastair Daivis	Redback	1	1	2	2	2	3	2	2	2	0	1	1	3	2	2	1	2	1	2	

Description	Files
Summary	Team HA - Skills ...- Team Skills.pdf
Distribution of skills	Team HA - Skills Assessment.pdf

| Learning Resources

Backend

- Pytest: https://docs.pytest.org/en/7.1.x/getting-started.html
- SQLAlchemy: https://docs.sqlalchemy.org/en/14/dialects/postgresql.html
- Flask :
    - https://flask.palletsprojects.com/en/2.2.x/
    - https://www.tutorialspoint.com/flask/index.htm
    - https://youtu.be/Z1RJmh_OqeA(Quick guide from setup, creating virtual env similar to the Developer.md , connecting to SQLAlchemy and the frontend)
    - https://github.com/realpython/discover-flask

Frontend

- React: https://reactjs.org/tutorial/tutorial.html
- Style components: https://www.robinwieruch.de/react-styled-components/
- Testing:
    - RTL Cheat sheet
    - RTL + Jest tutorial
    - Jest cheat sheet
- TypeScript:
    - TS Cheat sheet
    - TS tutorial
    - TS + React tutorial
- Storybooks:
    - Tutorial (start from step 2)
    - Official “how to write stories” guide

Accessing workflow related data

WorflowConfigurations data

WorkflowConfigurations class in workflow_configurations.py is a data class in the backend containing all the configurations needed for executing each stage in the whole workflow. This class is created and saved to the database when a new user is created.

The choices made by users from the frontend are captured in this class and then saved in the Users table in the database.

When new choices are added in the frontend, the WorkflowConfigurations class should be extended as well to accommodate the changes, so choices made by the user can be consistent throughout the whole application.

save_workflow and load_workflow functions in workflow_data_handler.py can be used in the backend to save and load the WorkflowConfigurations object to and from the database.

Dataset for machine learning and plotting

In the first stage of the whole workflow, featurized data will be created after the stage execution. This data is then also saved in the Users table in the database. Using save_featurized_data in workflow_data_handler.py will allow you to the dataframe to the database if there's any change to it.

To access the data for machine learning, or plotting, load_featurized_data in workflow_data_handler.py should be used to return the data.

Docker

Docker

Docker is a tool for managing isolated runtime environments (called "containers"). You can install Docker by following the Docker guide (linked below).

You can think of it as a combination of:

- A way to run virtual machines
  - But it's much more efficient
  - And makes it easier to access host machine resources, for example, network ports and filesystems
- A way to build custom virtual machines, just like `make` can be used to build binaries or applications
  - The build instructions are described by the `Dockerfile` file

Bear in mind, though, that **containers are not virtual machines** - that analogy is useful for describing an overview of what Docker does, but it is not accurate.

The two most important things you can do with Docker are:

- Build container **images**, for example, by running: `docker image build -t flask-docker .`
  - A container *image* is like a compiled program: it does nothing unless it is run
  - Unlike a program, though, it doesn't exist in an obvious way on the filesystem. The Docker runtime is responsible for managing images.
  - `-t flask-docker` tells Docker to "tag" the image with the string "flask-docker". You can refer to Docker objects (including images) by their hash, but naming them makes it easier.
  - The `.` tells Docker to use the Docker file in the current directory. You could provide a path to another directory instead.
- Run a container image, for example, by running `docker run -d -p 80:5000 flask-docker`
  - Running an image produces an object called a "container" - this is like a process, just as an image is like a compiled program.
  - The `-d` argument tells Docker to "detach" the container - to run it in the background and not show its output.
  - The `-p 80:5000` argument tells Docker to map port 80 on the host machine to port 5000 inside the container. If you're building a Flask application (which by default listens for requests on port 5000), this means that you would access the application by navigating to port 80 on the host machine.
  - `flask-docker` is the tag of the image we want to run.

Once a container is running, it is given a tag by the Docker runtime. This tag is separate from the tag of the image.

You can view running containers by running `docker ps`.

When you're finished with a container, you can kill it by running `docker kill <container-tag>`

For more information, you can go through the Docker getting started guide, or refer to their reference documentation.

Docker-Compose

An alternative to Docker approach detailed above
Docker-compose can automate the process of setting up & managing multiple containers. see Install Docker Compose

Before running the command below, create a .env file in this directory and add the following lines

```
POSTGRES_USER="db_user"
POSTGRES_PASS="somehardpassword"
PGA_USER="user@gmail.com"
PGA_PASS="anotherhardpassword"
```

Replace the text inside the inverted commas with the appropriate email/password/username

Then create a .pgpassfile in the "postgres-init" folder and add the following line

```
pg_db:5432:ha_db:username:password
```

Replacing username and password with the Postgres username and password chosen above

For security, ensure .env & .pgpassfile are added to .gitignore

Running `docker-compose up -d` in this folder will set up 3 containers with persistent storage

- Flask app as defined in **Dockerfile**, available at `127.0.0.1:5000`.
- Postgres database available at at `127.0.0.1:5432`.
- PGAdmin, a UI to easily manage the database `127.0.0.1:5050`. pgAdmin 4 Docs

Using a Docker Development Container

While it's a little unusual, you can also use Docker to set up a **development container**. This is a container that you would use to create an isolated environment for software development - for example, if your main machine is a Windows machine, but you want to use a Linux development environment.

Visual Studio Code includes some tools for this: https://code.visualstudio.com/docs/remote/containers

In this kind of setup, you would create a **separate container** for your development environment, and connect to it from Visual Studio Code.

From the VSCode terminal, you would be able to access the container via a shell just as you could on a normal Linux machine.

Inside the container, you could install Docker and develop the project as normal. However, there could be some issues with accessing web services hosted inside the development container. It may be simpler to use the **Windows Subsystem for Linux** instead: https://code.visualstudio.com/docs/remote/wsl

Git

Git

Introduction

I strongly recommend learning how to use Git from the command-line. You can find good resources for doing this on the internet, but one of the best places to start, in my opinion, is the Atlassian Git tutorial.

The initial part of this tutorial is mainly about setting up a repository, which we have already done and might not be so useful to you. I would recommend starting with Saving Changes, which introduces the "three trees" of Git. Understanding how those trees work together is fundamental to understanding how Git works and how to use it effectively.

A second, also excellent resource is Thoughtbot's Git Tutorial, which comes with written tutorials, recorded videos, and example repositories. In particular, their end-to-end feature example very closely matches how I like to use Git.

Branching strategy

Branches are key to using Git effectively. While the tutorials above give some good advice on using them, every team has a slightly different branching strategy, so I'll mention some good guidelines here:

- **Always use a branch**. Branches keep you safe by making sure you can get back to a given state of the repository. If you're about to do something risky (for example, a `git amend` or `git rebase` command), make a branch first so that if it goes wrong, you can check out the "before" version of your code and undo the mistake.
- **Do not modify the history of shared branches**. If you are sharing work on a branch with other developers, using a command like `git commit --amend` or `git rebase` can cause your version of the branch to become different from the version that the other developers have.
- Adopt a c**ommon strategy for naming branches**. I like to use the following formula:
  - Use a slash (`/`) to separate different parts of the branch name
  - A brief description of the **kind of work** being done on the branch (`feat` for new feature development, `style` for code style changes, `doc` for adding documentation, `tests` for adding unit tests, and so on)
  - A short description of the **actual work** being done on the branch, for example `feat/backend-featurizer-list-endpoint`

Resources

- The official git tutorial
- Atlassian's git cheatsheet
- Dangit, Git!?! is a good resource for figuring out how to fix a mistake you've made using Git

Tools

While your IDE may have built-in Git support, I recommend using a stand-alone tool. They tend to be more stable and provide a good overview of the state of the repository.

I use Sublime Merge, but other popular tools include:

- SourceTree, by Atlassian
- Tower
- tig, a command-line UI

Again though, I strongly recommend becoming familiar with Git's command-line interface. Often the GUI interfaces are convenient but hide information that can help you actually understand what they are doing, and knowing the command-line interface will help you use them more effectively.

Related to the above, if you are not yet familiar with a command-line editor, I would strongly recommend learning one.

By far, the most popular of these are Vim and Emacs, but both of these have quite steep learning curves. If you are looking for a command-line editor to get started with, the easiest is probably `nano`, which comes installed on most MacOS and Linux machines (I'm not sure about Windows, sorry!).

For those of you who are already confident with Git, some other tools you might find interesting/useful include:

- Interactive Rebase Tool, a more convenient interface for using `git rebase -i`.
- Trail, a tool I developed for rebasing feature branches on top of an amended commit after responding to code review feedback.

GitHub

GitHub

Introduction

GitHub fills two key roles in our project:

- Code hosting server: it acts as a Git server from which we can push and pull commits, and by doing so share our work
- Code review platform: it acts as a way for us to share feedback on each other's work, facilitating quality control, collaboration, and knowledge distribution.

If you're familiar with Git already, most of GitHub should not be new to you. The exception to this is GitHub's **Pull Request workflow**, which covers the process of merging changes from a feature branch into the main branch.

With this in mind, I recommend reading GitHub's documentation on this topic.

Tools

- GitHub's command-line interface is very good
- GitHub Desktop is a desktop application, which you may find more convenient than using GitHub's website.

| PyTest

Pytest is a python testing framework that is used to write a small, readable test that can be scaled to a complex test such as testing API or database and UI. A testing framework is used to make sure our code behaves as we expect and ensure any changes to the code won't cause a regression in the program. The reason we use Pytest is because of the eases of use of the library. Pytest can be installed onto your machine by manual installment of Pytest packages or installed through the requirements.txt, which is located in the `HA-2022-SM2\src\backend`

Install command:

```
python -m pip install pytest
```

```
HA-2022-SM2\src\backend > python pip install -r requirements.txt
```

In this project, Pytest is used to test several parts of the project. For example, it is used to test if the databases can be used to read and write. It is also used to test the program if the workflow is able to be updated, if a user is able to log in/register using test data, and many more. Pytest is integrated into our Continuous integration, where we test any code with Pytest before reviewing a pull request to make sure the code is able to work as expected.

For more info on how to learn Pytest, the following links are recommended:

- https://docs.pytest.org/en/7.2.x/getting-started.html#get-started
- https://www.tutorialspoint.com/pytest/index.htm