Individual Journal

ICT30001 - Information Technology Project

Class: Fri 13:00 DT2.2 - **Tutor**: Dr. Nguyen Phuong Anh

Name: Trac Duc Anh Luong - **ID**: 103488117 - **Group 2**

Journal Entry #1

The first week was our introduction to the project. For the lecture, we covered materials related to understanding, establishing and defining the problem or opportunity within the organisational context. Our team was formed, including the following positions: 1 AI Engineer, 1 Data Engineer, 1 System Analyst, and 2 Software Engineers. The lecturer proposed the topic of tree mapping, using computer vision to detect trees and their health status. The topic was later changed to House map, using CV to detect houses & buildings in urban areas and detect yearly changes. In this topic, the team roles match well, and our task was to find a client for this project.

Journal Entry #2

The lecture covered leadership, teamwork, people management, and stakeholder identification this week. Our group found a client - NTT e-MOI JSC, a subsidiary company of NTT East, a well-established telecommunications organisation from Japan. The company's contact point is Xuan Dung Luu, a software engineer. For internal messaging, the group used Messenger and Zalo to message and keep our client and mentor updated. Our group agreed to use ClickUp as the primary project management tool. As I was trained to use ClickUp at work, I was able to do a quick knowledge transfer session with other teammates on how to use the tool and created a hierarchy to control the folder structure:

- Tasks
- Wiki
- Resource
- Meeting Notes
- Docs
- Asm

Journal Entry #3

This week's lecture revolves around problem definition, scoping, and Business Requirements. The individual project description statement assignment was due at the end of the week. The assignment goal was to determine the system scope, investigate the current situation and relevant issues, and analyse the feasibility. Requirements, including functional and non-functional, had to be defined. For this assignment, my brief description of the work covered data collection & processing, AI & smart vision development, training models, generate map data on coordinates for the AI team. UI design; system integration & testing; map generation & visualisation on map for SE team. Both teams will provide training & support for the client. The main requirement is to detect changes in buildings and houses in Hanoi city in terms of shape and size. Two big problems arose: Where can we find datasets for training and testing? How can we gather yearly data in the same area? The tasks for the software team are to design the database for yearly housing data storage; to determine Which tech stack can be used, considering the short development time, which the solution had to be developed fast with quality standards that can

satisfy the client; Use third-parties API (Google) for the map and figure a way to visualise the acquired data.

Journal Entry #4

The lecture for this week was related to Project Management and designing holistic solutions from a multidisciplinary perspective. The first group assignment is the Initial Project Proposal and plan. The assignment covered project management, project methodology, requirements prioritisation, requirements management and documentation, research and evaluation of alternative solutions. This week, the project's scope changed to visualising data on images. We need to figure out a way to display images as maps and how to visualise metadata on these images. The solution to last week's problems: The team found a dataset from Wuhan University using data from an urban area in New Zealand. The AI team used Google Earth SDK to get yearly images and experimented with data from Hanoi Starlake. In terms of task progress, the designed schema follows:

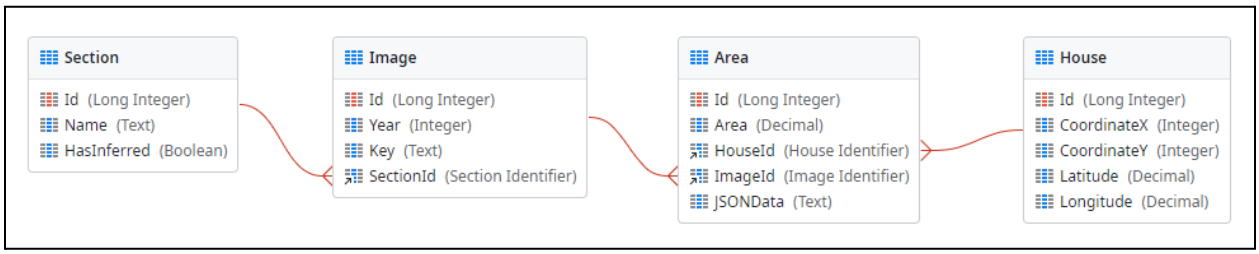

Figure 1: Database schema

For the relationships, 1 Section had many Images, one image for each year. 1 Image had many Houses, and 1 House had many Images. For this, I had to create a many-to-many relationship and decided to use the Area entity to normalise the schema and store Area information as well as the shape of the house. For the tech stack to be used, the team decided to use OutSystems, a low-code development platform I have experience using. It is the platform that our client, NTT e-MOI, is using. In the research process, I found a way to use JavaScript to get the position X & Y on the web canvas in the map div using Event. The problem was that while we could use this position to draw the centre of each house, how could we store and visualise the multiple bounding shapes that the houses had?

Journal Entry #5

This week, we learned about CMS and user documentation. Regarding task progress, the Software team researched and found a solution for displaying map data. We would display centre points from X & Y stored in the database for each house. Then, we draw a circle div, add the id of the div & an on-click event listener to that div. Finally, we append the circle to the Map div. For the on-click behaviour, the Accordion item in the list on the right of the screen will expand to display yearly area data associated with that House Id. For this solution, the Image has to be a

background URL for the div. The user needed to zoom in and out of the Div like a map using buttons. Therefore, we used JavaScript transform. We then wrapped the Map div in a container div to hide overflow & create a map-like interface. The year dropdown was added to refresh the Data Action to load images and data from each year. The problem for this week is: Where can we store images and get the image URLs? We planned to use cloud storage like AWS S3 & use the method GetPresignedUrl from the AWS SDK. We would first need to install the SDK. For the following tasks, we needed to research JavaScript libraries to display bounding shapes on the web. We then need to send a meeting invitation to the client and prepare materials (slides, docs, scripts) for the following week's presentation with the client.

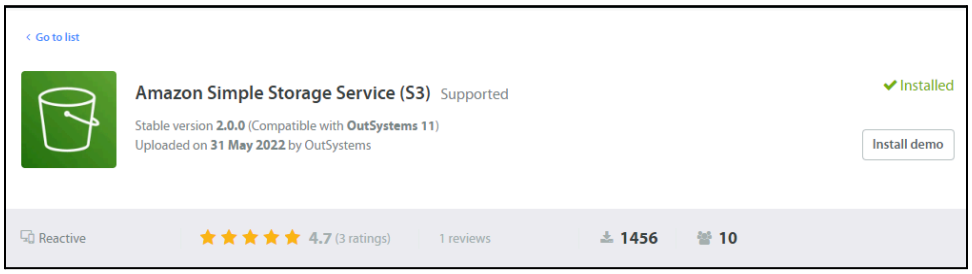

Figure 2: AWS S3 installed and integrated

Journal Entry #6

For this week, we studied deriving business value and conducted the Client Progress presentations assignment. The assignment focused on communication with stakeholders and change management issues. The team conducted a client progress meeting, showing the current progress with the project, Gantt Chart & planned future work. We actively discussed the project with the client in the meeting and asked for comments and recommendations. The client commented that the project description & objectives are clear, the overall progress is up to date with our Gantt Chart, and we should prioritise essential tasks like the AI model and the front end. Regarding the recommendations, we were advised to keep in touch regularly for updates and support. The app's UI/UX can be improved, and we can create a theme for the app in OutSystems. We must create a data visualisation dashboard for new requirements and add functionalities to export PDF reports of all sections and Area data in Excel, CSV, or JSON format. This week, I have found a JavaScript library - P5.js to visualise the bounding box. To integrate the library in OutSystems, I found a sample community module to implement the library in our app. Initially, we can only draw quadrilateral shapes. Then, I found the function vertex() in the library to draw complex bounding shapes.

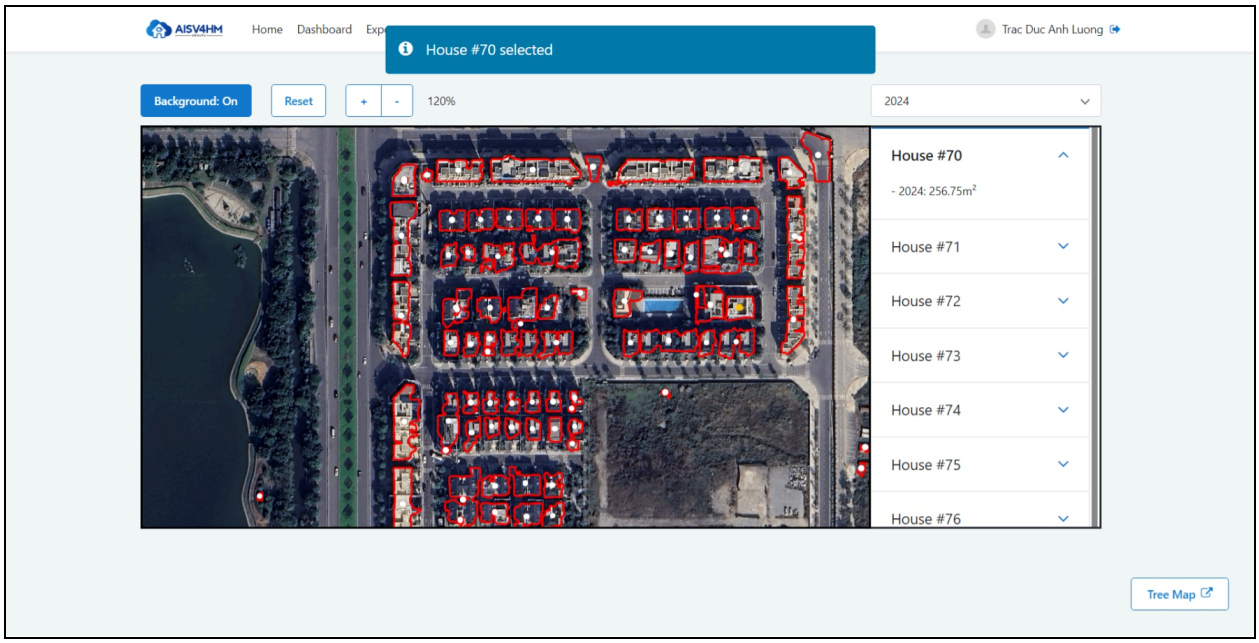

Figure 3: P5.js integration for bounding box drawing

For this method to work, we must store the JSON X & Y of shapes in text format as planned. Next, we planned to test the app with data from the AI team to find any potential bugs and necessary adjustments.

Journal Entry #7

This week, we started to develop a screen to create sections and upload associated metadata & yearly images. As the inputs were from the user, some validations are needed, including:

- Unique section name
- Unique key (file name)
- Different year for each key (image file name)
- Image type (jpg, png) & meta-data type (tif, tiff)
- The image would be uploaded to S3. Therefore, we used the section name as a prefix/sub-directory. Consequently, they have to be unique.
- Validate if the section has already existed on S3.

A question arises: How can the AI team know the current image for which year? We devised a solution to rename the file to the year associated with the image. Upon upload, the S3PutObject method initially got connection timeout for files larger than 1MB. The solution was to increase the module timeout from the default 10s to 120s. A new task was to expose an API for the AI team to write data. For security, we implemented a basic authentication, ensuring the consumer had to log in using a username and password for access. Initially, the SE team only exposed single CRUD actions for the AI team to write each record through the API. This was slow and inefficient; therefore, I devised a plan to expose action and insert bulk data instead. The problem with this approach was how we could know the current House existed in another year's image to

insert into DB for the AI team. For this, the SE team created a method to get the HouseId from the lat & lon the AI team sent us. As the image is taken from different angles, there will be disturbance up to a certain precision point. Therefore, we had to create an extra param to send in the precision point, round the lat, lon received from the API and the House lat, lon in the DB up to that param, using ROUND() in SQL. After testing using AI team data, the endpoint's performance is good (1.5s to 2s) to insert 152 Houses & Areas. House boundings are off due to a mismatch in the image ratio since the web ratio is fixed. The following tasks were making the interface responsive for different image sizes and calculating the respective ratios when drawing centre points & bounding shapes.

Journal Entry #8

This week's main task was to fix the app's responsiveness. The purpose was to make images of all ratios correctly displayed on the web canvas in all screen sizes. Our approach was:

- Set CSS style for URL background so that it will fit the div by default
- Get the width of the Map div at the OnReady event using Javascript
- Get the image size from S3 Presigned Url
- Set height of Map div = ImgHeight * DivWidth / ImgWidth
- Use Map width and height to set the size of the canvas
- Calculate the ratio to draw centre points and bonding shapes using Ratio = ImgWidth / DivWidth
- Set the height of the Accordion List of house data to the same height as the Map div
- Create a default class when there is no data returned from the Data Action

→ For this approach, the turbulence on the front end is less than 1px, and this number was accepted by our mentor & client.

To view all area data associated with a house in different years, we created a House Block, put it in the Accordion item and displayed it upon Accordion expansion. The todos for next week were to research a method to implement a list nested in a list to improve the performance of the Map screen, as well as reformat the code and work on test cases.

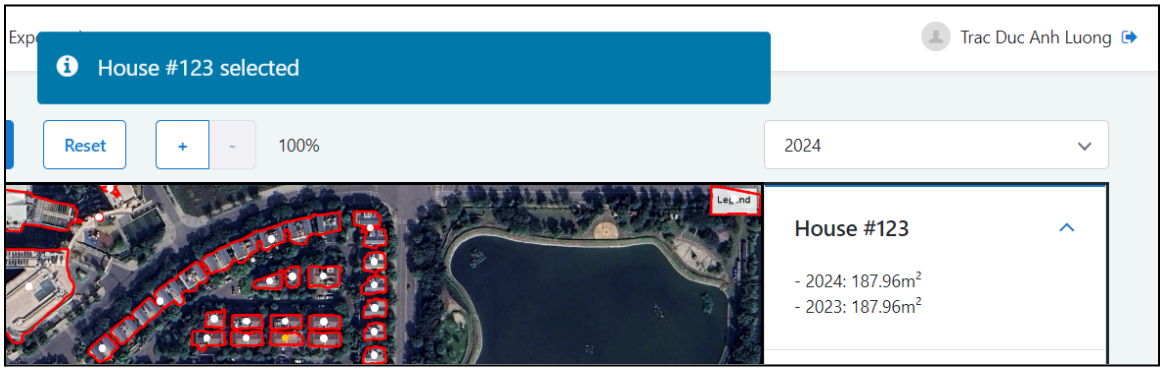

Figure 4: Accordion item displayed area data of a house in multiple years

Journal Entry #9

This week, the lecture covered software testing and quality management. The new requirement is to detect trees & yearly growth of tree areas. The tree detection will be cloned into another module with separate endpoints and DB, ensuring new changes will not affect the current module. Once both detections run smoothly, they will be integrated into the same module. Even though the two modules are different, they still use the same S3 bucket. Therefore, we have to check if the sub-directory (annotated as the section name) exists alongside checking the section name in the DB. For completed tasks, the SE team and I improved the app's performance using SQL and grouped Server Actions. Instead of looping through OutSystems' Delete Action, SQL was used to delete Houses in a nested statement when pruning data of a Section. In the SectionCreate screen, we grouped the SectionCreate, upload meta-data & image actions into a single Server Call instead of multiple server calls. We found a bug in the exposed API and OutSystems CRUD action during the testing phase. The CreateHouse action automatically rounds the latitude and longitude of the House when inserted in the DB. It can only store eight decimal digits from the 14 digits received from the API. The affected part was that after inserting the identical 152 houses in different years, if the ROUND() function is set to 6 decimal digits, it will create one new House. If the ROUND() function is set to 7 decimal digits, it will create 22 new Houses. The solution is to use SQL to insert House, directly rounding and truncating the lat & lon in the statement to ensure they are not automatically rounded.

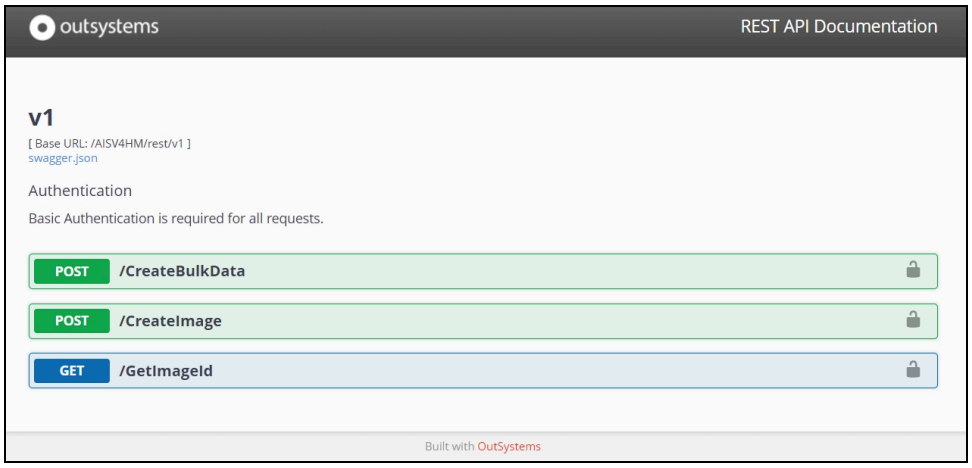

Figure 5: REST API created for the AI team to insert housing data

Journal Entry #10

This week's lecture focused on report writing. With the existing data collected from many urban areas in Hanoi, the team decided to create a dashboard for the project. The data action will be filtered by year to push data to the charts, while a year picker similar to the Map screen will be used to refresh that data action. For the components of the interface, there would be two charts to display the total house count & area sum by each section, two counters to display the total house count & area sum of all sections, and a data table to show each record, which can be later used for the PDF report that we were planning to export next week. The user could select the current

year from the year picker to view data in terms of functionalities. They could also choose data points from 1 of 2 charts to view corresponding data in the table. Finally, we would show a message to the user when the data is filtered and an option to view the entire data table.

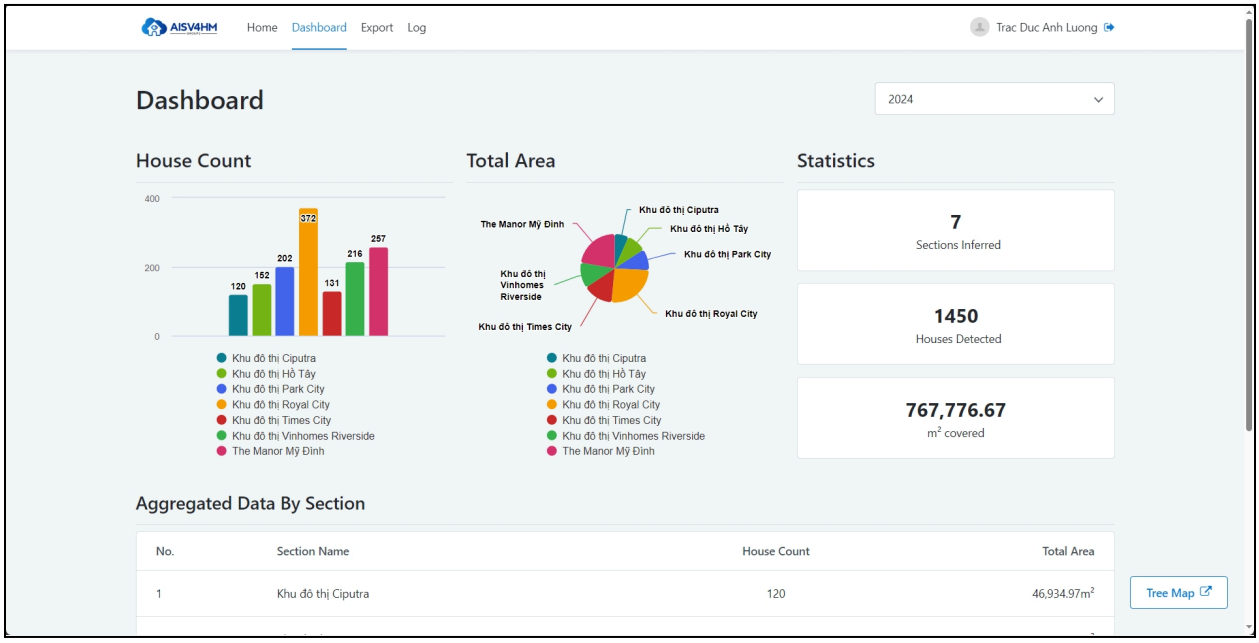

Figure 6: Interactive dashboard to display housing data in the database

Journal Entry #11

The week hosted a presentation skills workshop, including the reporting and presenting findings, preparing us for the final presentation. For the export functionality, the SE team and I researched and successfully exported data in all required file types: Excel, CSV, and JSON, as well as a PDF report containing aggregated data from each section. For the PDF report, we created a screen using the print layout. The screen would have the following information: exporter, export date, and data table. We used the PrintToPDF server action to print the wanted page. Then, we used JavaScript to get the offset between the server and the client to write the export date in Date Time format. The Excel, CSV, and JSON data files used the same query to join all four tables in the database. For JSON, we had to deserialise the ShapeItem List before serialising it again and converting it to a binary data file. The file encoding was set to UTF-8 to correctly display Vietnamese characters in the section name and S3 image key.

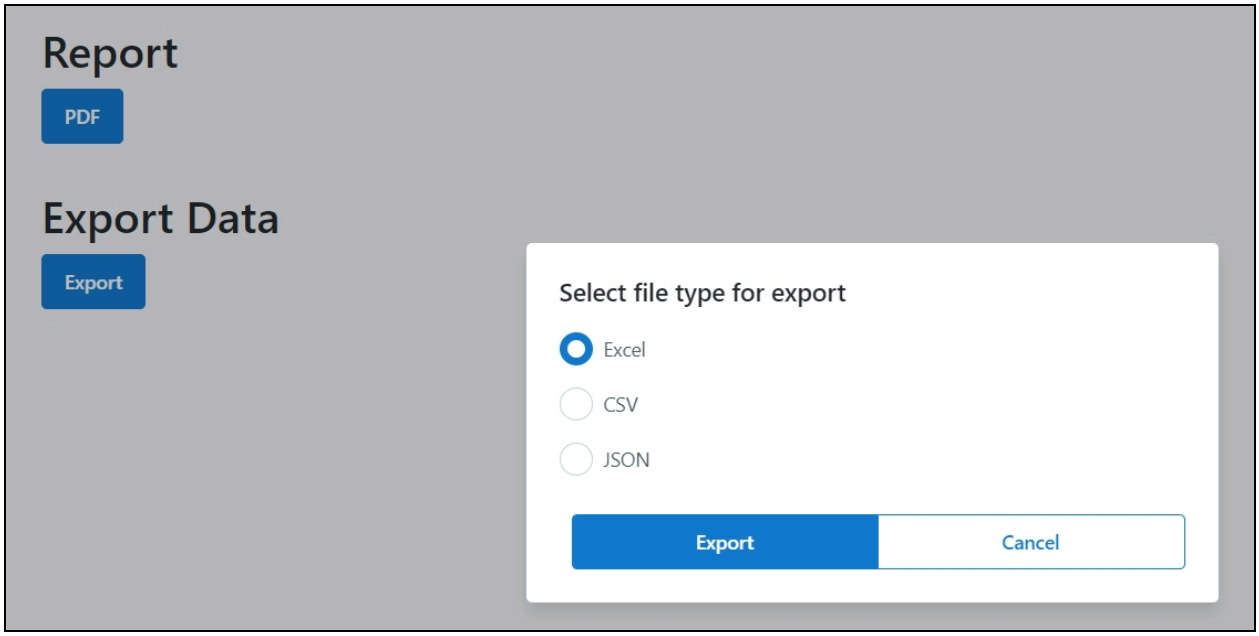

Figure 7: File export options

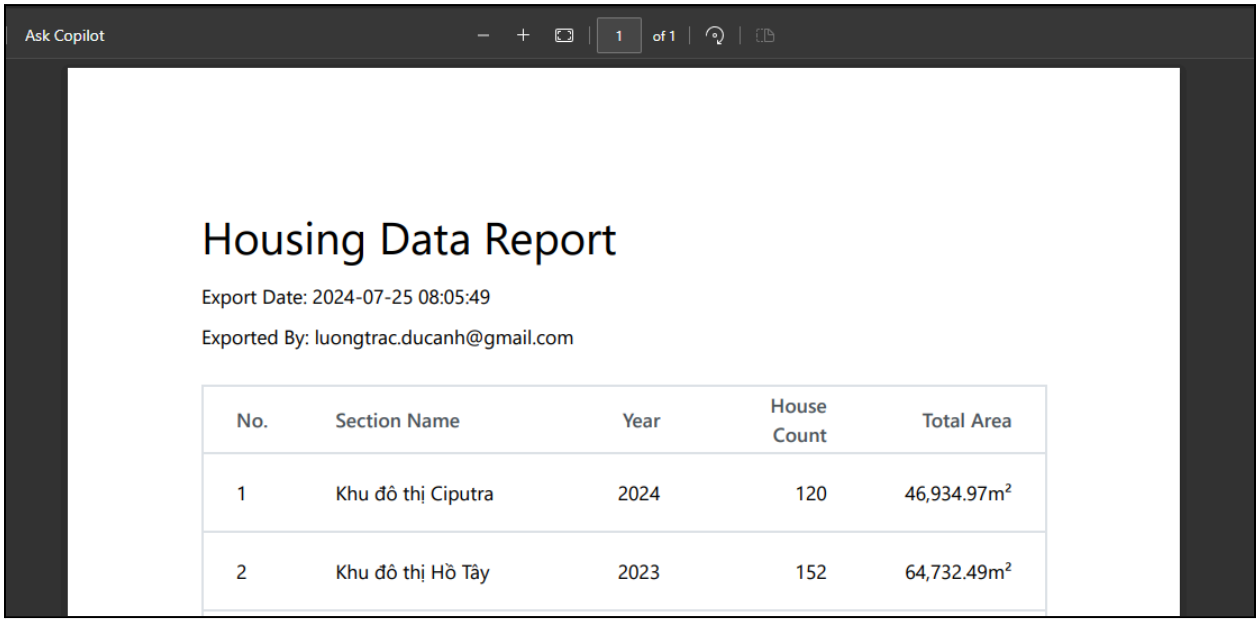

Figure 8: PDF report exported

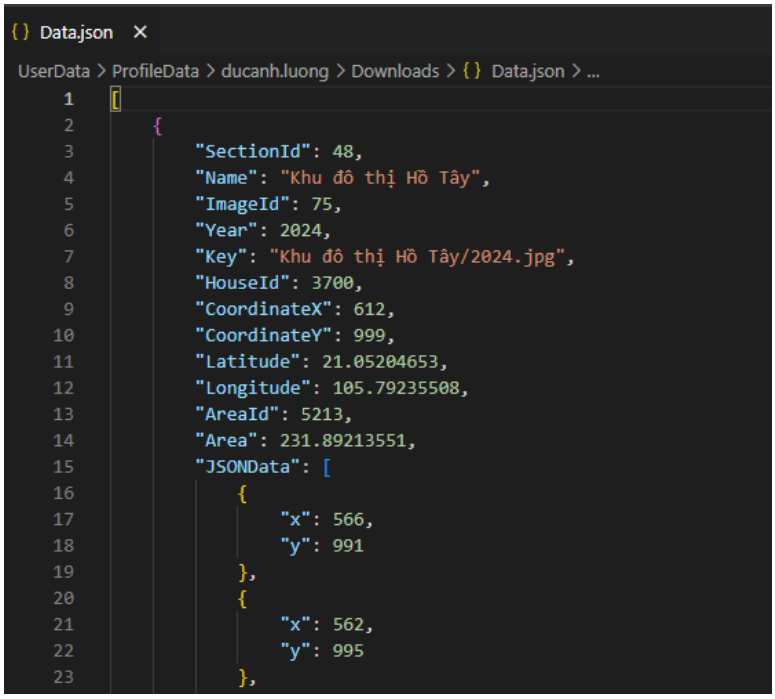

Figure 9: JSON Data exported

Journal Entry #12

For this week, the team discussed final project submissions and client signoff and practised our presentations. For new features, I have added a screen to monitor logs in real time, adding dependencies to the platform logs and getting data from the Log_General table. Crucial actions such as calling and getting responses from the EC2 API, section data from the exposed REST API, exceptions, and file exporting will add the LogMessage action and post the corresponding messages.

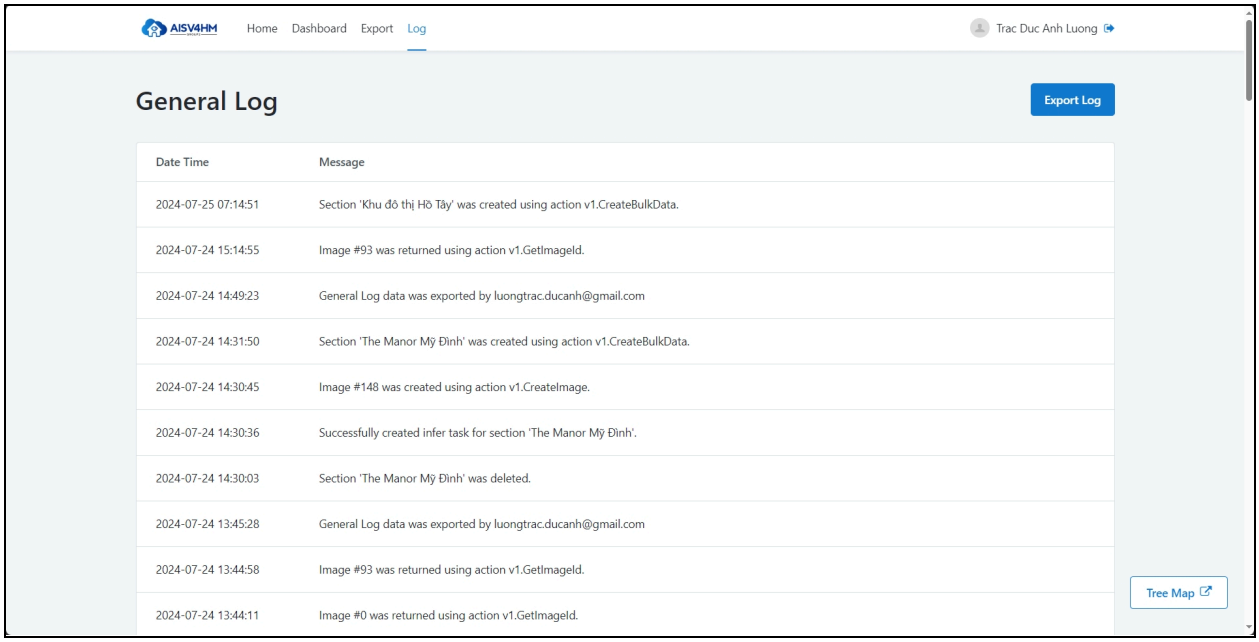

Figure 10: General Log from REST and CRUD actions

For the presentation, the team's client and mentor are happy with the final product.
The stakeholders commented:

1. **Innovative**: The team successfully created an AI-powered web application that was innovative successfully.
2. **High application**: The system could be easily expanded and/or integrated with new and existing systems, providing tangible value to the client. This flexibility ensures the longevity and adaptability of the software in a rapidly evolving technological landscape.
3. **Enhanced UI**: The UI was significantly enhanced from the previous client meeting progress, with responsive design and loading animations for heavy data-processing actions.

From this project, I have learned a lot about working in teams and creating industry-standard software that is:

1. **Usability**: Designing intuitive interfaces and user flows that cater to diverse user needs and preferences.
2. **Maintainability**: Implementing clean, well-documented code and following best practices to ensure the software can be easily understood and updated by future developers.
3. **Extensibility**: Architecting the application with modularity, allowing for future enhancements and feature additions without compromising the existing functionality.

My technical proficiency has increased due to this project, but it has also given me a better grasp of the software development lifecycle and the significance of matching technology solutions with organisational goals.