LBVT roadmap

technical documentation

Team: 2024-SP2-24

Name: Yuta Kasuga

The Kiet Ly (Sam)

Contents

[1. Program outline 3](#_Toc168359034)

[2. Web page 3](#_Toc168359035)

[How to use the roadmap 4](#_Toc168359036)

[Welcome building. 5](#_Toc168359037)

[Course building 6](#_Toc168359038)

[Elective building 9](#_Toc168359039)

[Industry building 11](#_Toc168359040)

[Alumni building 13](#_Toc168359041)

[Camera 14](#_Toc168359042)

[Styling (SCSS) 15](#_Toc168359043)

[3. Database 15](#_Toc168359044)

[XML 16](#_Toc168359045)

[Json 17](#_Toc168359046)

[Python 17](#_Toc168359047)

[4. 3D modeling 17](#_Toc168359048)

# Program outline

This product is 3D web-design interactive program roadmap for Environmental Science (LBVT) for University of South Australia (Uni SA). Users can find course information, prerequisite for the course and the connection between futures’ one, elective information for the program, and the related YouTube videos in Uni SA’s official account.

Our team use these technologies for this project.

Web page:

* HTML, CSS, JavaScript
* React.js, three.js, tailwind.css

Database:

* XML, Json, python

3D modeling:

* gltf, blend

Software:

VSCode: Coding, launch the project

Blender: Handling 3D modeling

# Web page

This product is web-based road map therefore, we used mainly use coding technology which we learn in the courses except 3D modeling.

Users can check the information with related building with clicking the building on the roadmap. When user click the building, new window will be popped up on. Due to our client want to display course connections on the map, we use html technology not react.js for this part. The product has several types of buildings such as:

* Welcome: Displaying YouTube videos which related program.
* Course: Displaying course information.
* Elective: Display all type of electives’ course.
* Industry: Displaying partner companies of Uni SA.
* Alumni: Displaying alumni’s information

For displaying UI sections, we did not use react.js due to our clients wanting to display course connection in roadmap, and we thought changing page with react.js was not suitable for this project.

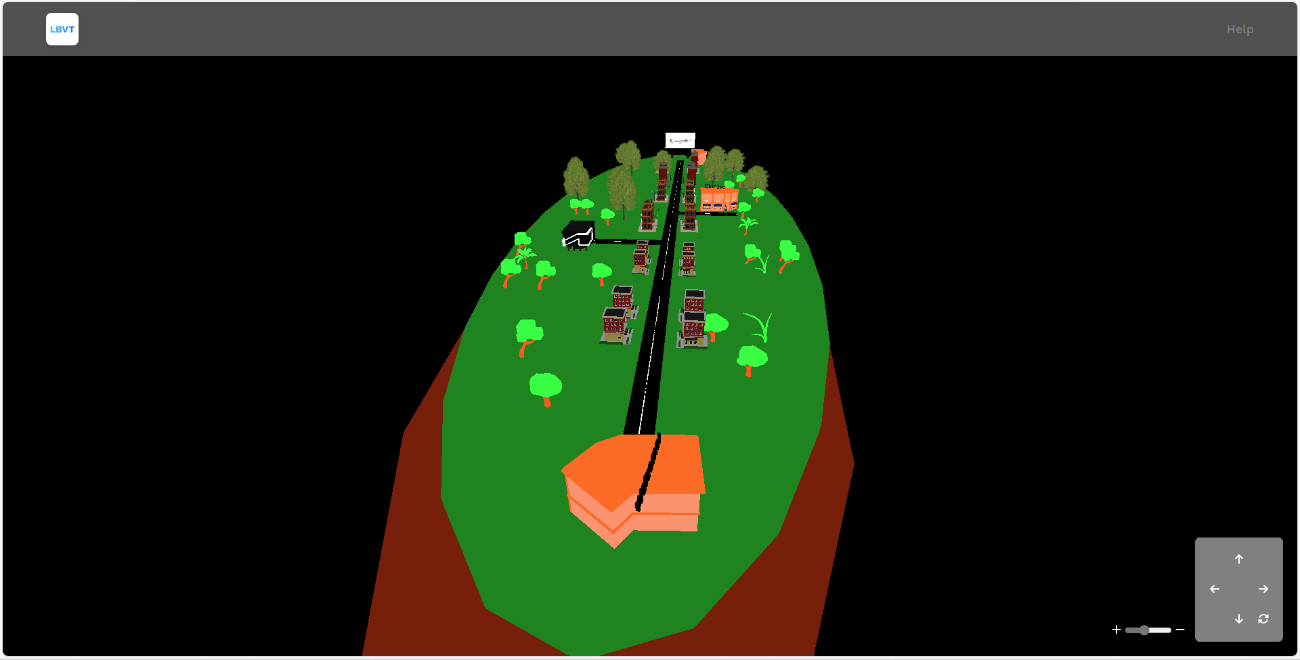


Figure: Program Roadmap

## How to use the roadmap

* To launch the application. Enter “npm install” command in VS Code([Visual Studio Code - Code Editing. Redefined](https://code.visualstudio.com/))’s terminal to download necessary information. To launch the project, enter “npm run dev” command in the terminal the press “o” and enter key.
* To move your location, hold your right click down then move your mouse. or click arrow buttons which are right bottom of the screen.
* To move your camera, hold your left click down then move your mouse. Or slide a +/- bar which are right bottom of the screen to modifying the height.
* To reset camera, click the circle button which is on right bottom corner.
* To zoom in, scrolling up your mouse and scrolling down for zoom out.
* When you click the building, the related information will be popped up.
* To change the roadmap, click the link in the help page or enter “/program=lbvt” or “/?program=iboe” after the localhost::XXXX. (ex. localhost::5173/program=lbvt)

## Welcome building.

This building has responsibility for welcome message and displaying videos which related the program for perspective students. Therefore, they can understand what they will study during university life, campus information, and program director’s message. The videos are from Uni SA’s official YouTube channel.

For coding part, we use function in Home.jsx. To generate new window, we used openWindow(), and adding CSS and contents with Javascript. The data is from Json file for LBVT or IBOE course in assets folder. For much detail, please also check the database part in this document.

Functions:

* displayWelcomeUI()

Display information for welcome UI. The coding is mainly used html, css, and Javascript for clicking event.

Parameter: N/A

Return: N/A



Figure: welcome building on the roadmap

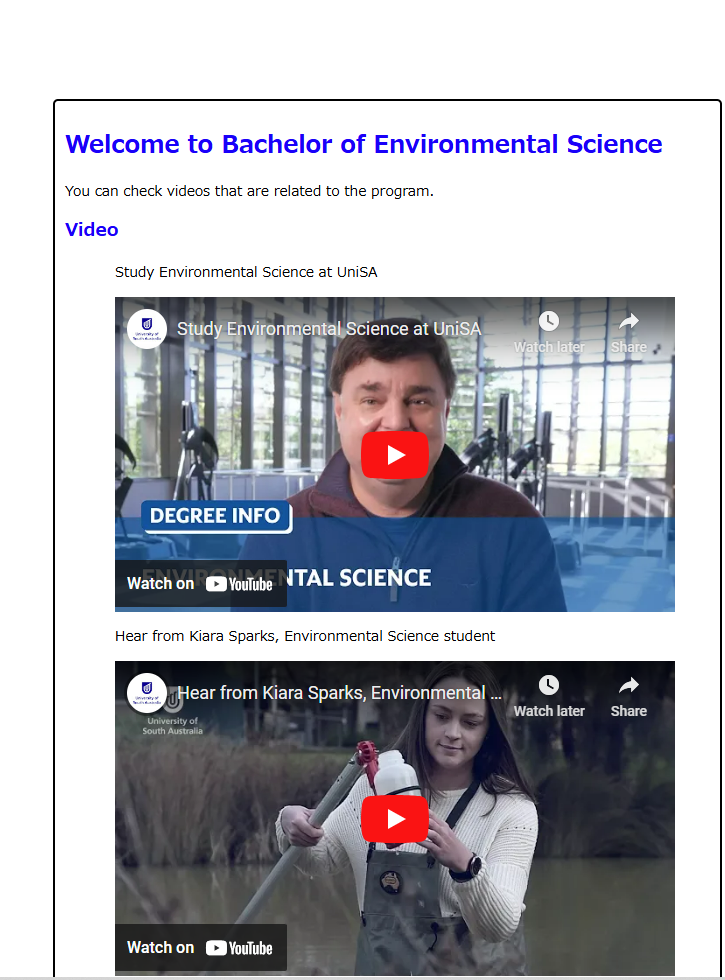


Figure: window for welcome building

## Course building

For imaging journey, we have multiple buildings and each building has each course information of the program. There are 24 course buildings on the map and each building has one course information. (Note: For LVBT program, there are 23 courses, so one building is not having UI to display.)

If the course has prerequisite for the future courses, the related building will be highlighted with green. In course UI, user can check course information such as course name with link to Uni SA course page, course id, course aim, and so on. User can check the related courses in the page as well. For coding part, we use functions in Home.jsx.

Functions:

* onMouseClick(event)

Detect the building which has clicked.

* displayCourseUI(evt, connectedCourses, id)

Display course window.

Parameter:

evt: All general courses’ information as Object

connectedCourses: Prerequisite courses in future as Set

id: Building id on the roadmap

* getPrerequisite(courseID)

Return prerequisite for the course course as object.

Parameter:

courseID: course ID as string (ex:GEOE2026)

Return:

Course information as Object.

* findFuturePrerequisite(courseId, courses)

Find and return all related course in future.

Parameter:

courseID: course ID as string (ex:GEOE2026)

courses: all courses in general course as Object

Return:

All courses which will be required as prerequisite in the future as Set.

* findConnection(courseId, courses)

Find and return all related buildings ID in future course.

Parameter:

courseID: course ID as string (ex:GEOE2026)

courses: all courses in general course as Object

Return:

All building IDs in roadmap for future class as Set

* updateLine(buildingObject, buildings, connectedCourses)

Highlight the buildings which are related course which user clicked.

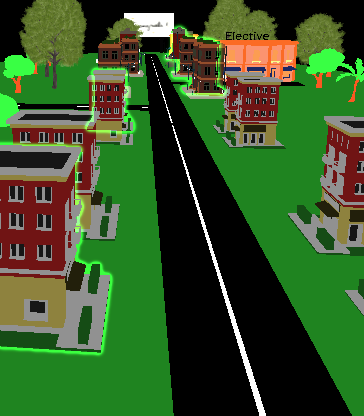


Figure: course buildings with highlighted

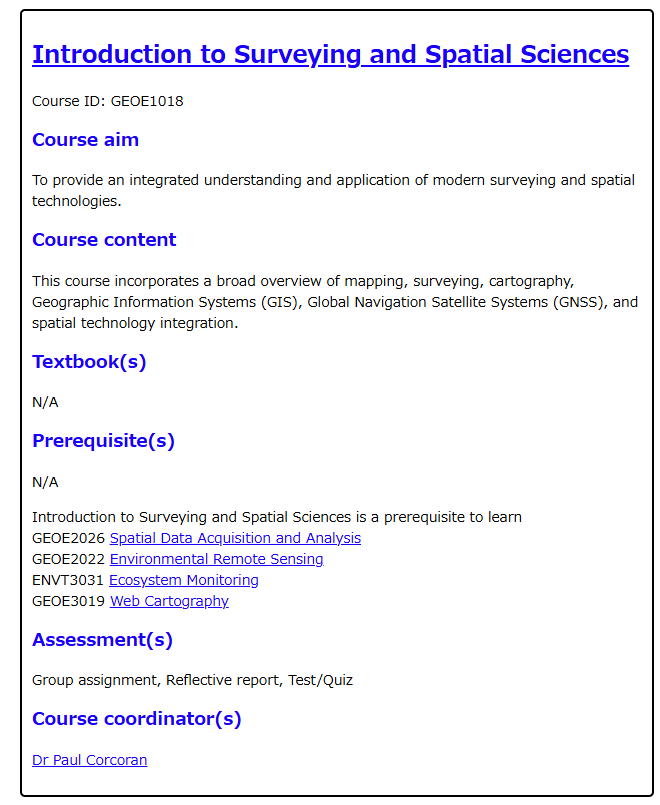


Figure: Course UI for GEOE1018

## Elective building

This building has responsibility for displaying elective information. When user click the building, new window will be popped up with electives’ information. Every elective section’s information such as undergraduate elective, and environmental elective will be displayed.

For coding, we use functions in Home.jsx.

Functions:

* onMouseClick(event)

Detect the building which has clicked.

* displayElectiveUI()

Create new window with elective information for the program.

Parameter:N/A

Return: N/A

* addElectiveUI(courseUI, electiveJson)

Add each elective course information to window.open() in displayElectiveUI() as HTML format.

Parameter:

courseUI: Variable which store window.open() in displayElectiveUI()

electiveJson: A course information as Object

Return: N/A

* getPrerequisite(courseID)

Return prerequisite course information as object.

Parameter:

courseID: course ID as string (ex:GEOE2026)

Return:

Course information as Object.



Figure: elective building on the roadmap

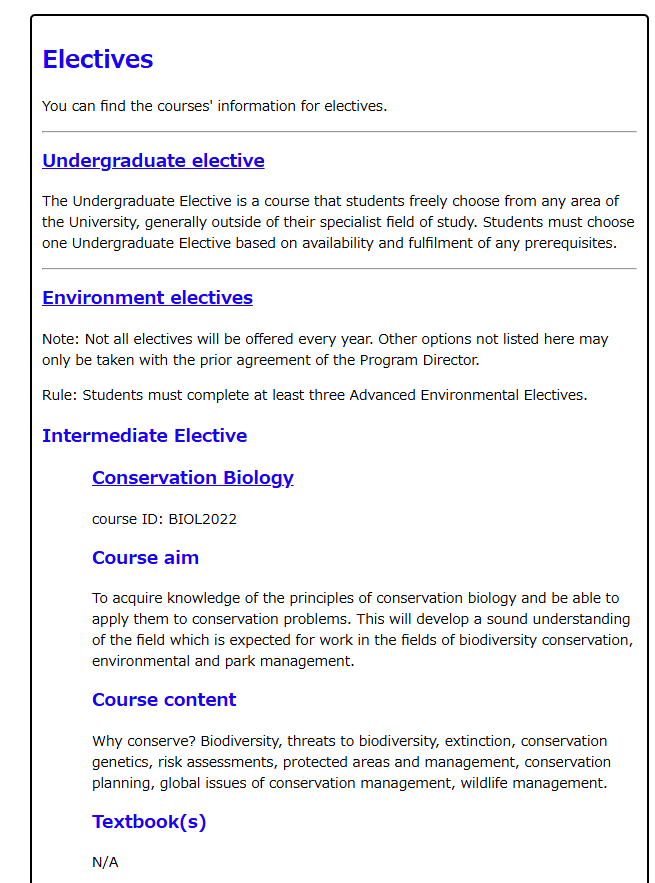


Figure: Elective page for LBVT program

## Industry building

This building has responsibility for displaying partner companies of Uni SA for the related industry. Due to focusing on to perspective students for the stakeholder, we have not implemented the link for Career service of Uni SA because they don’t have student id to login the Uni SA’s service.

For coding, we use functions in Home.jsx.

Functions:

* onMouseClick(event)

Detect the building which has clicked.

* displayIndustryUI()

Create new window to display partner companies for the program.

Parameter: N/A

Return: N/A

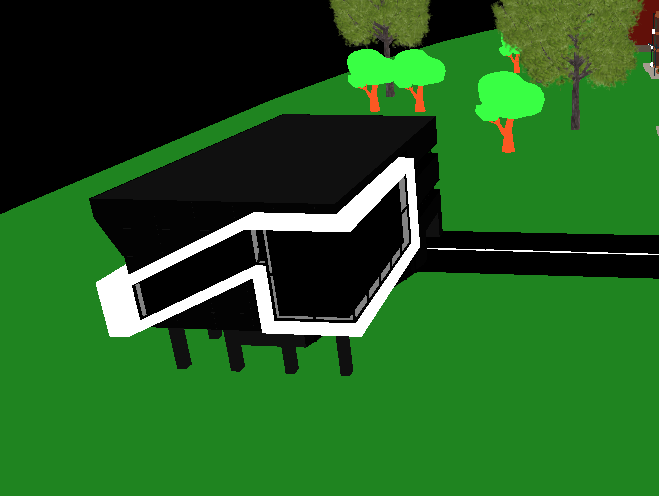


Figure: industry building on the roadmap

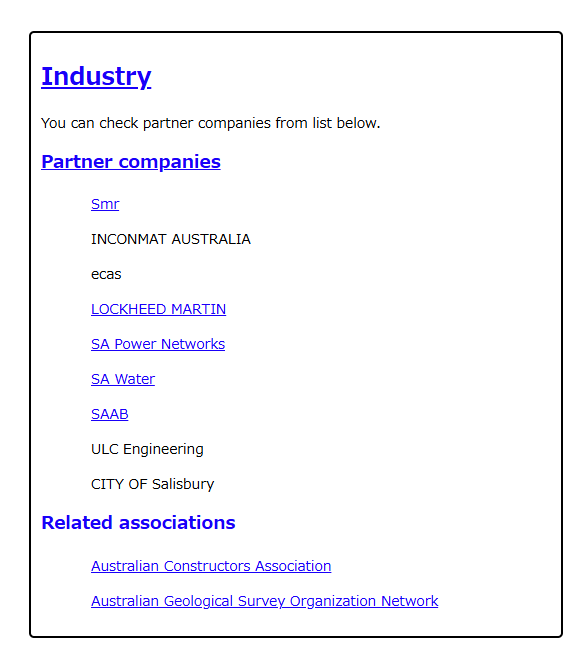


Figure: industry page for LBVT program

## Alumni building

This building has responsibility for displaying the link to alumni’s information.

For coding, we use functions in Home.jsx.

Functions:

* onMouseClick(event)

Detect the building which has clicked.

* displayAlumniUI()

Create new window to display the link to alumni’s information.

Parameter: N/A

Return: N/A

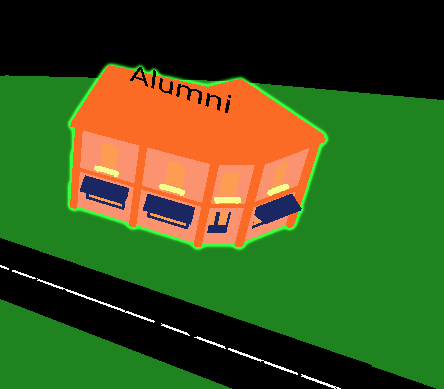


Figure: Alumni building on the roadmap.



Figure: Alumni page

## Camera

We implement buttons and slide bar to move cameras to handling 3D image smoothly.

How to use buttons:

* Arrow buttons (→, ↓, ←,　↑):

Move the camera’s position slightly to the direction

* Rotate button on the right bottom of the corner:

Reset the camera’s position to initial place.

For the coding part, Buttons.tsx, cameraSlider.jsx, ControlPanel.tsx are handling the camera.

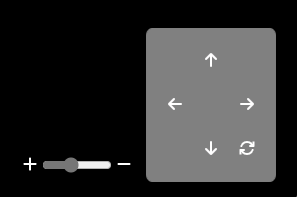


Figure: buttons and slide bar for camera

## Styling (SCSS)

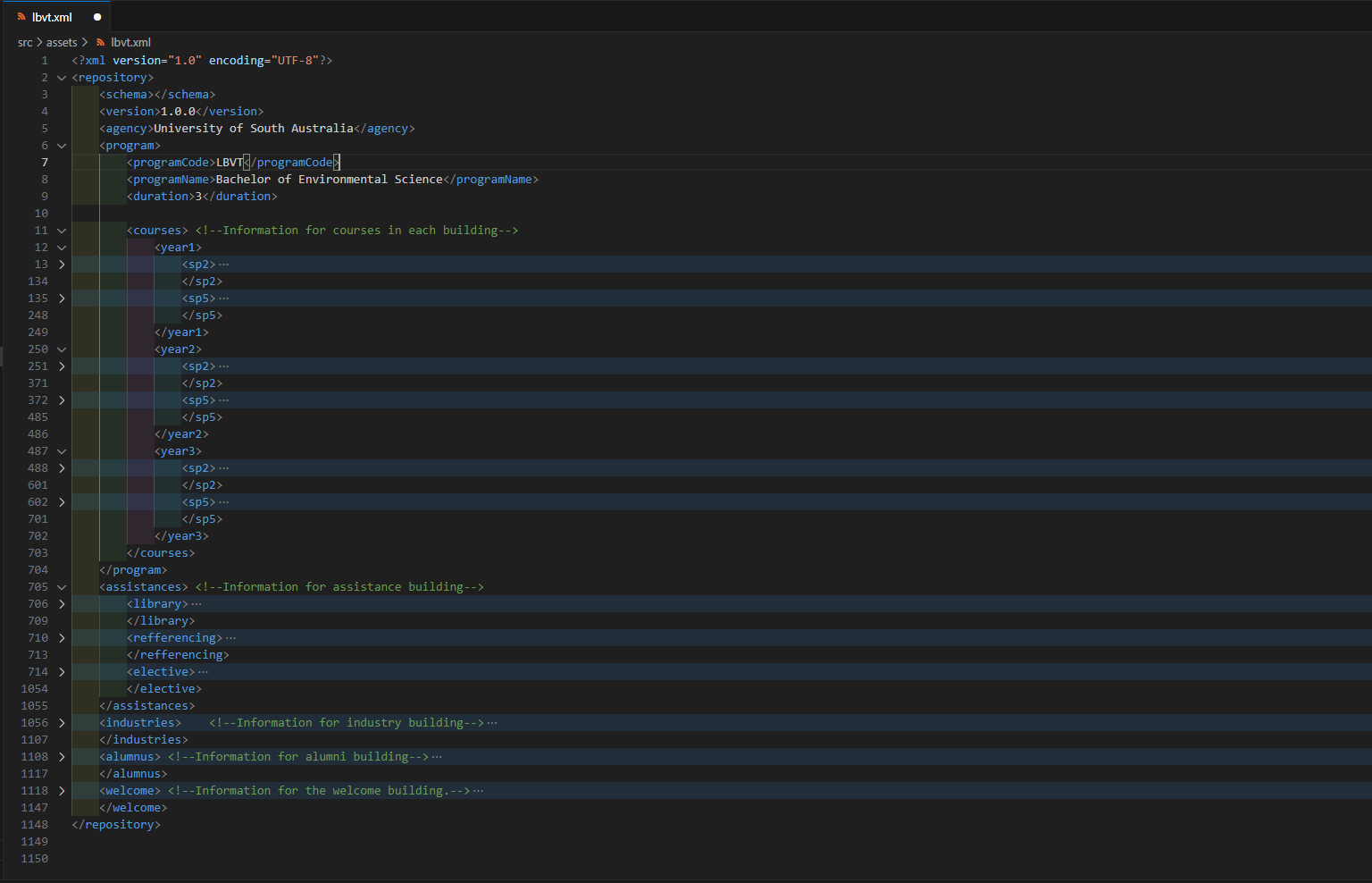
Except UI for clicking event, our products use tailwind.css that are in “src/styles”, and “src/styles/components”. Each react component such as Buttons.tsx, cameraSlider.jsx, ControlPanel.tsx has each “.scss” file in “src/styles/components” folder and having coding for styling. To activate each .scss file, we need to write a code to activate in the \_components.scss in “src/styles” folder.

# Database

We used the XML and Json file for the database. XML is actual database for the program and Json is used for displaying information in web page. To convert XML to Json file, you can use “xmlToJson.py” in “src/utils/xmlToJson.py”.

## XML

XML has every information for the program, such as program name, duration of the program, every course information for each semester, elective information and so on. The xml file for LBVT is in “src/assets/lbvt.xml” and “src/assets/iboe.xml” .



**Structure:**

<program> -> About program information.

<courses> -> About courses information which contain each year and each semester.

<yearX>-> Containing courses in year X

<spX> -> Containing courses in semester X

<course> -> Containing course information.

<assistance> -> About assistance information mainly for Elective.

<library> -> For the link to the library. (Not used current version.)

<referencing> -> For the link to the referencing. (Not used current version.)

<elective> -> Containing every type of elective.

<electiveCourses> -> containing several type of elective (ex. Environmental, undergraduate)

<course> -> Containing course information.

<industries> -> About the partner companies which are on Uni SA webpage.

<partner> -> Containing company information

<associations> -> Related association for the program.

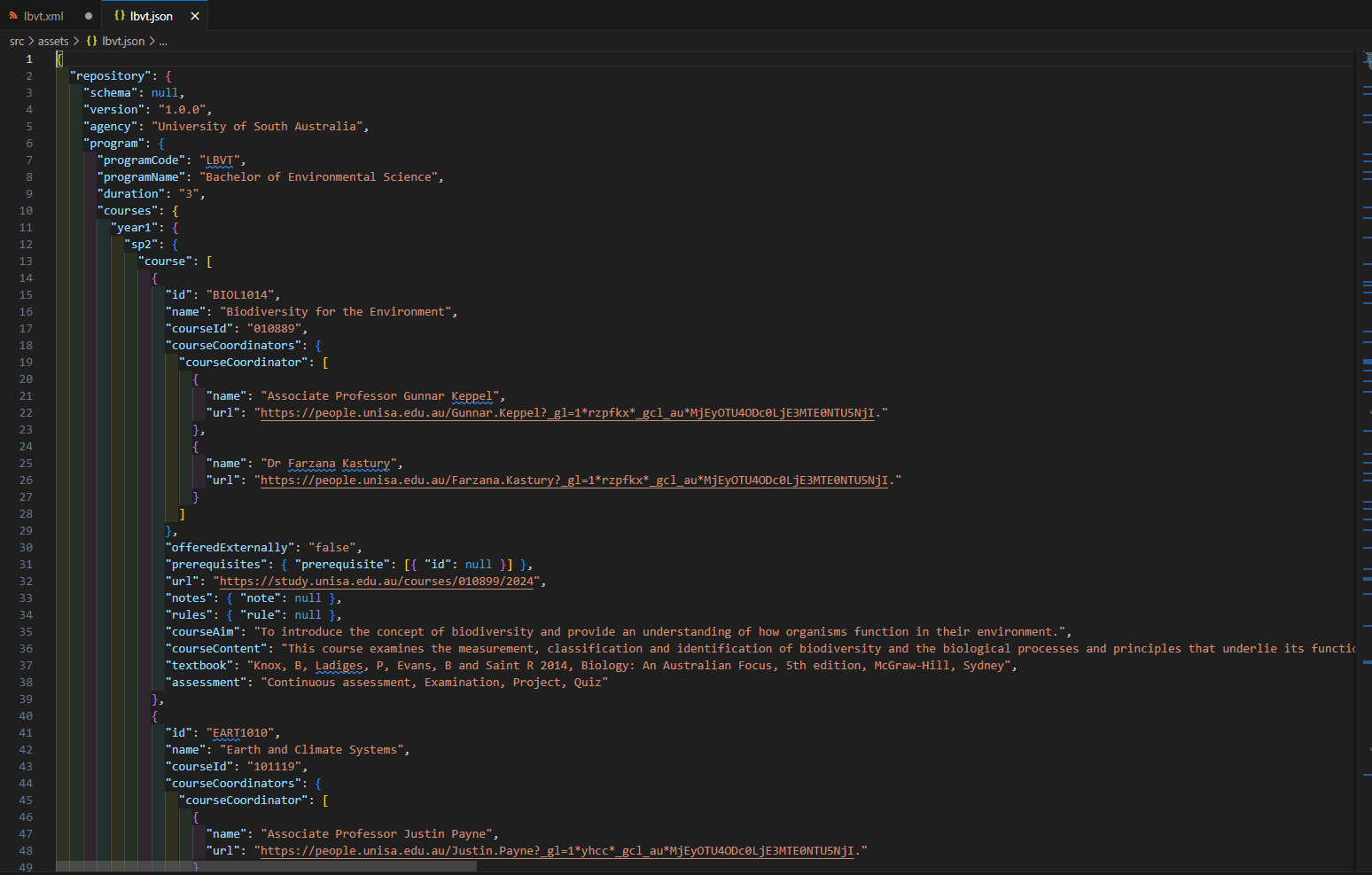
<Alumnus> -> About alumni’s information

<welcome> -> Containing YouTube videos’ link which are related to the program.

<videos> -> Containing YouTube videos’ URL

## Json

Json file is used for displaying information in web page. Json file is converted from XML file by “xmlToJson.py” in “src/utils/xmlToJson.py”. Structure is same as XML file. Json file is in “src/assets/lbvt.json” and “src/assets/iboe.json”.



## Python

In case of adding new program for the project for future improvement, we have python file to convert XML to Json. You need to set paths of XML file (line 14 in xmlToJson.py) and where to generate Json file (line 21). xmlToJson.py” is in “src/utils/xmlToJson.py”

# 3D modeling

For creating 3D modeling, we use Blender ([blender.org - Home of the Blender project - Free and Open 3D Creation Software](https://www.blender.org/)) which can modify and generate 3D modeling. It is able to convert the 3D modeling from .blend file to .gltf/.glb file after creating the model. 3D modeling files (.blend and .gltf/.glb) are stored in “src/assets”. Some 3D assets are downloaded from Sketchfab ([Newsfeed - Sketchfab](https://sketchfab.com/feed)). For the detail, please check the “license.txt” in the project.

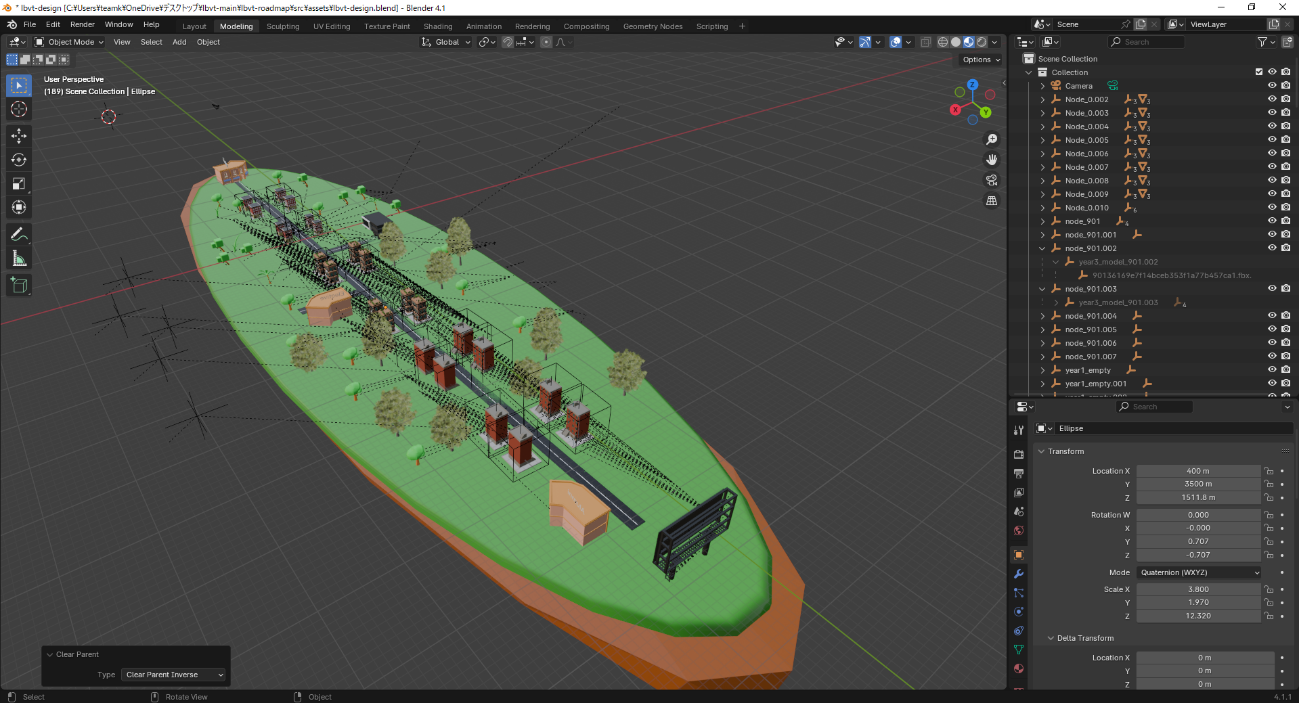


Figure: 3D modeling in Blender

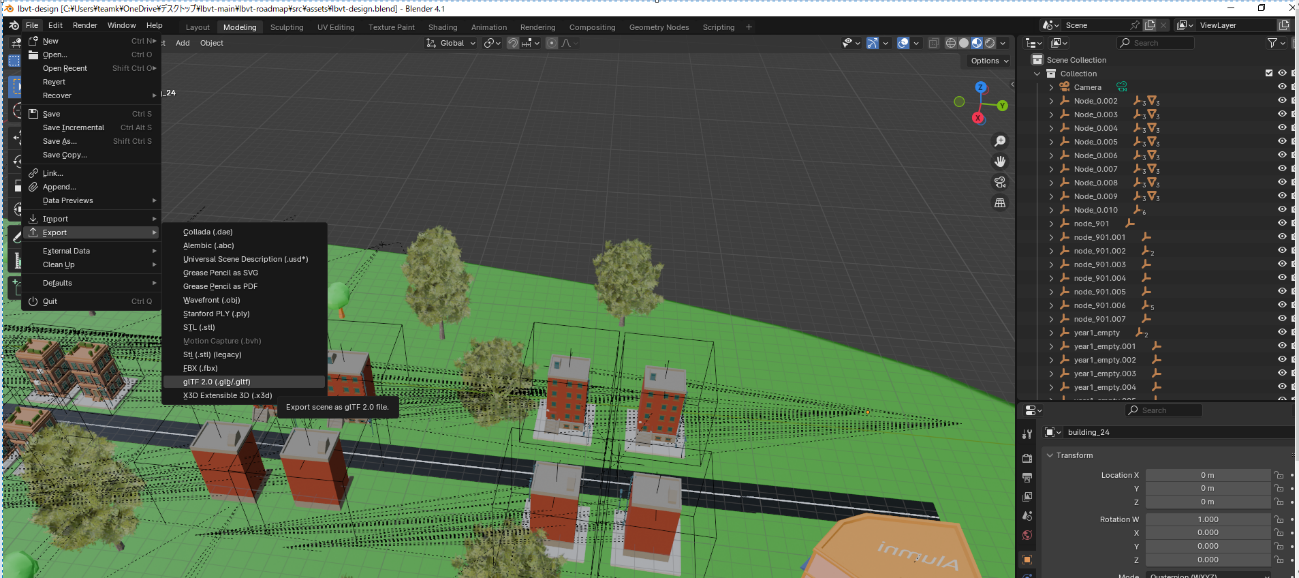


Figure: Exporting .glb/.gltf file in Blender

Each building has name (ex. Building\_1, grate\_hall) for handling onClick() function in Home.jsx.