Project Presentation and Demonstration

Group no: 04

Members: Lucas Qin
Natalie Huang
Mona Ma
Yichen Song

SWINBURNE UNIVERSITY OF TECHNOLOGY

Acknowledgement of Country

We respectfully acknowledge the Wurundjeri People of the Kulin Nation, who are the Traditional Owners of the land on which Swinburne's Australian campuses are located in Melbourne's east and outer-east, and pay our respect to their Elders past, present and emerging.

We are honoured to recognise our connection to Wurundjeri Country, history, culture, and spirituality through these locations, and strive to ensure that we operate in a manner that respects and honours the Elders and Ancestors of these lands.

We also respectfully acknowledge Swinburne's Aboriginal and Torres Strait Islander staff, students, alumni, partners and visitors.

We also acknowledge and respect the Traditional Owners of lands across Australia, their Elders, Ancestors, cultures, and heritage, and recognise the continuing sovereignties of all Aboriginal and Torres Strait Islander Nations.

Overview

- Background
- Scope and Objectives
- Outcomes
- Design concept 1 Search
- Design concept 2 Community
- Compatibility of Design
- Code demonstration
- Website demonstration

Project Background

From the past to the present, countless touching works written by talented **female composers**. However,it's hard to find even a single female work on most of today's core repertoire lists.

The Corelia project aims to **re-promote underestimated** and even **overlooked female composers and their work**, thereby pushing the classical music cultural scene in a more diverse direction.

Project Scope and Objectives

- Build an online database for female composers and their works
- Allow users to quickly find a specific composer or piece
- Display page for each composer
- A blog section for recommendations
- A forum that allows users to communicate

Project Outcome/Deliverables

- A NoSQL database
- An online web site to display the data
- A management platform to update data

Project deliverables will consist of multiple web page designs, a web prototype, project test data, and the final product (including database, web front and back, and servers).

Design 1 Search function

Specifications

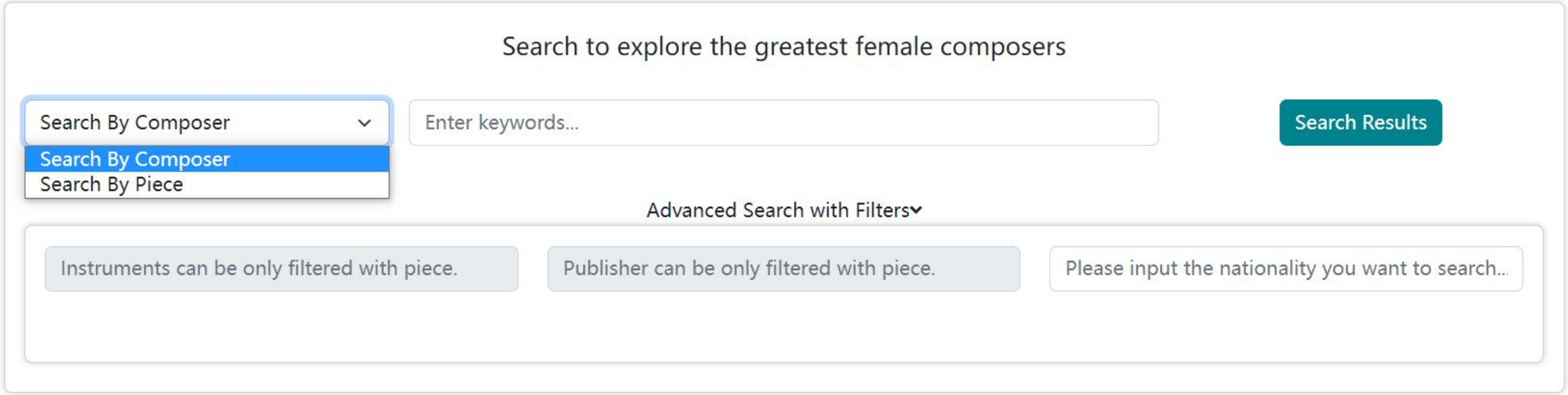
- Search for pieces or composers
- Advanced search function: filter
- Fuzzy search
- Error Correction
- Search prediction

Constrains

- Require high performance server
- Require machine learning

Design 1 Development Search function

- Search by COMPOSER or PIECE
- Apply filters: nationality, instruments, publisher…
- Fuzzy search



Design 2 Share and Community

Specifications

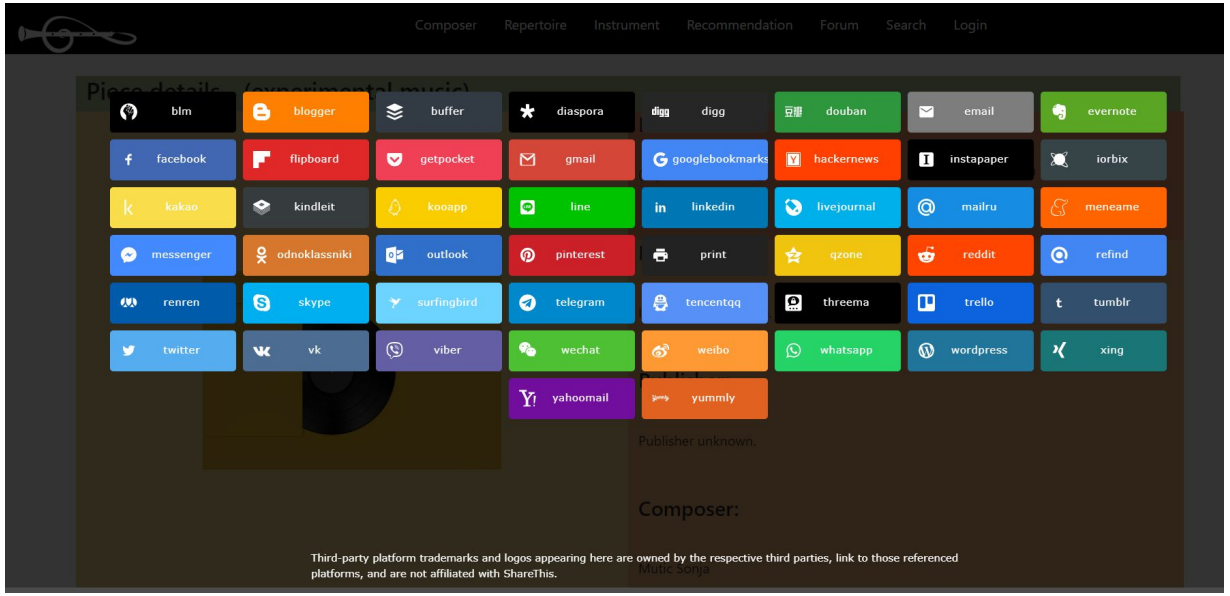
- Share all composers and pieces to social networks such as facebook
- Leave comments to pieces
- Third-party platform for communities
- Corelia official account content display on website on live
- Join in groups, post, comment, like, share or browse

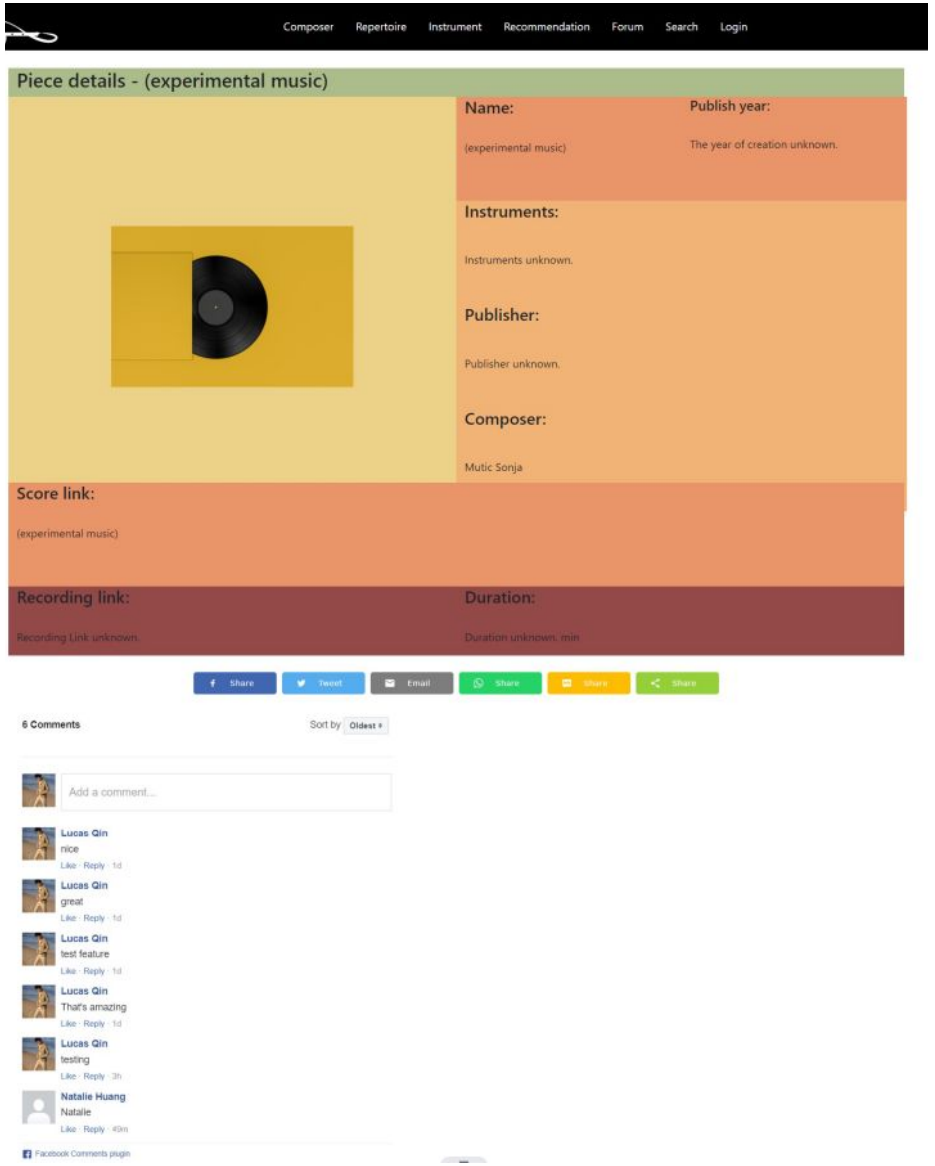
Constrains

- Require accounts for those social networks
- Third-party portals could change constantly and need maintenance

Design 2 Development Share and Community

- Share any page
- Comment any page
- Share to most popular social medias

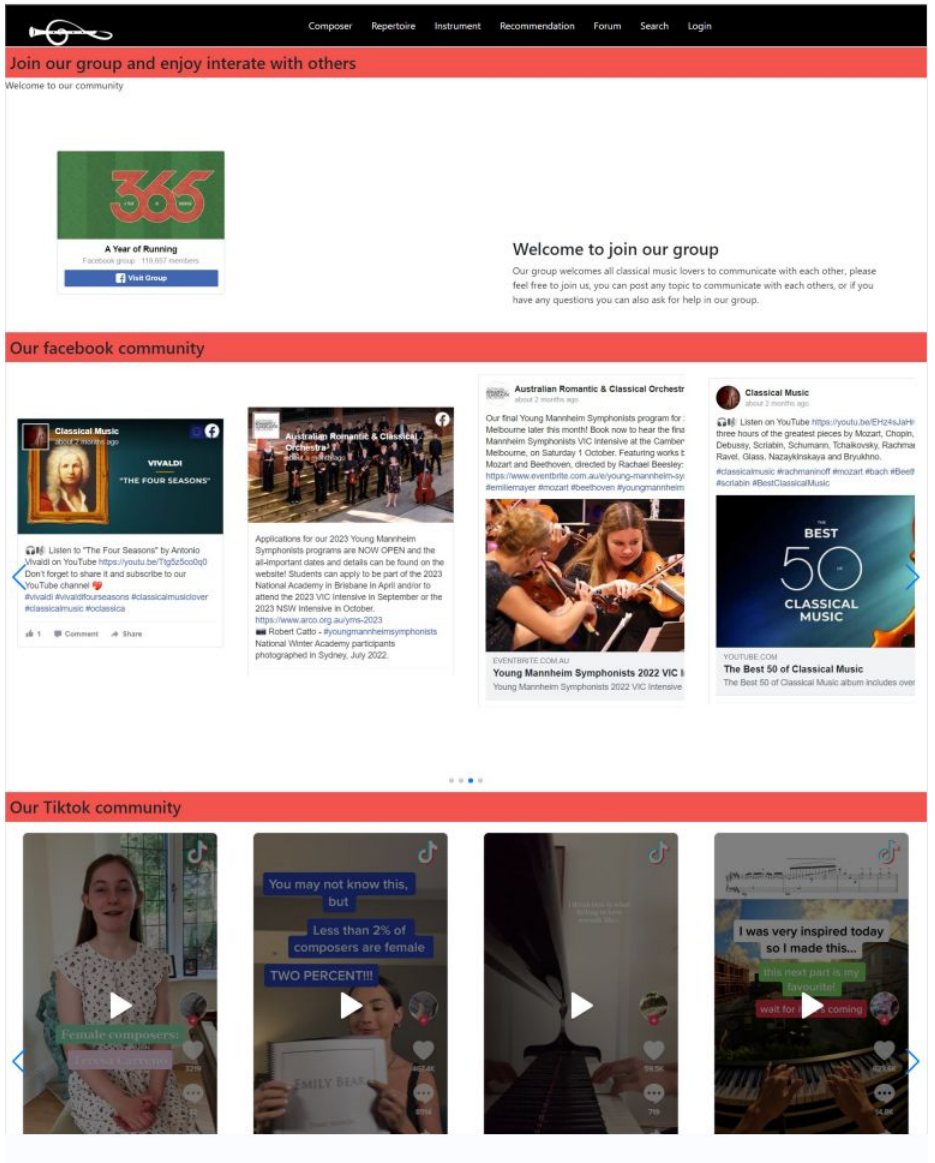




SWINBURNE UNIVERSITY OF TECHNOLOGY

Design 2 Development Share and Community

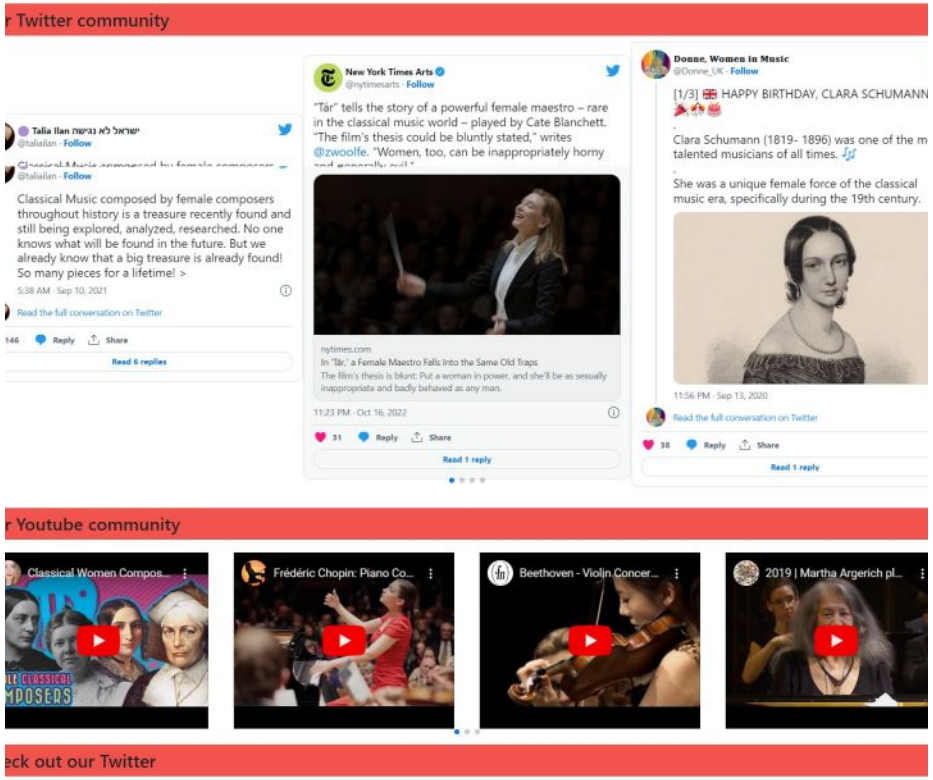
- Join community group
- Display featured composer’s social medias
- Share, like, comment and watch
- Facebook, Tiktok



SWINBURNE UNIVERSITY OF TECHNOLOGY

Design 2 Development Share and Community

- Twitter, Youtube
- Share, like, comment and watch
- Corelia Official account content go live



SWINBURNE UNIVERSITY OF TECHNOLOGY

Compatibility of Design

The client requirements which we completed.

1. Search function with filter
2. Forum and communities
3. Share contents
4. Composers display and profiles
5. Pieces display and details
6. User authentication
7. Recommendation Page
8. Instruments Page
9. Admin account:
    - Modify piece list
    - Modify composer list
    - Modify recommendation list

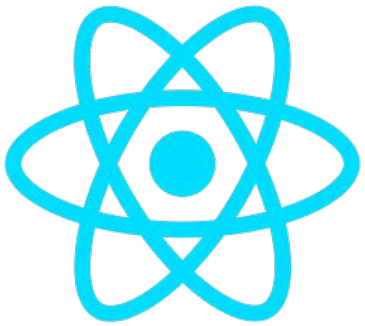
Code Demonstration

Technical Stack: MERN

Database

Backend

Frontend



React JS

Bootstrap

Testing

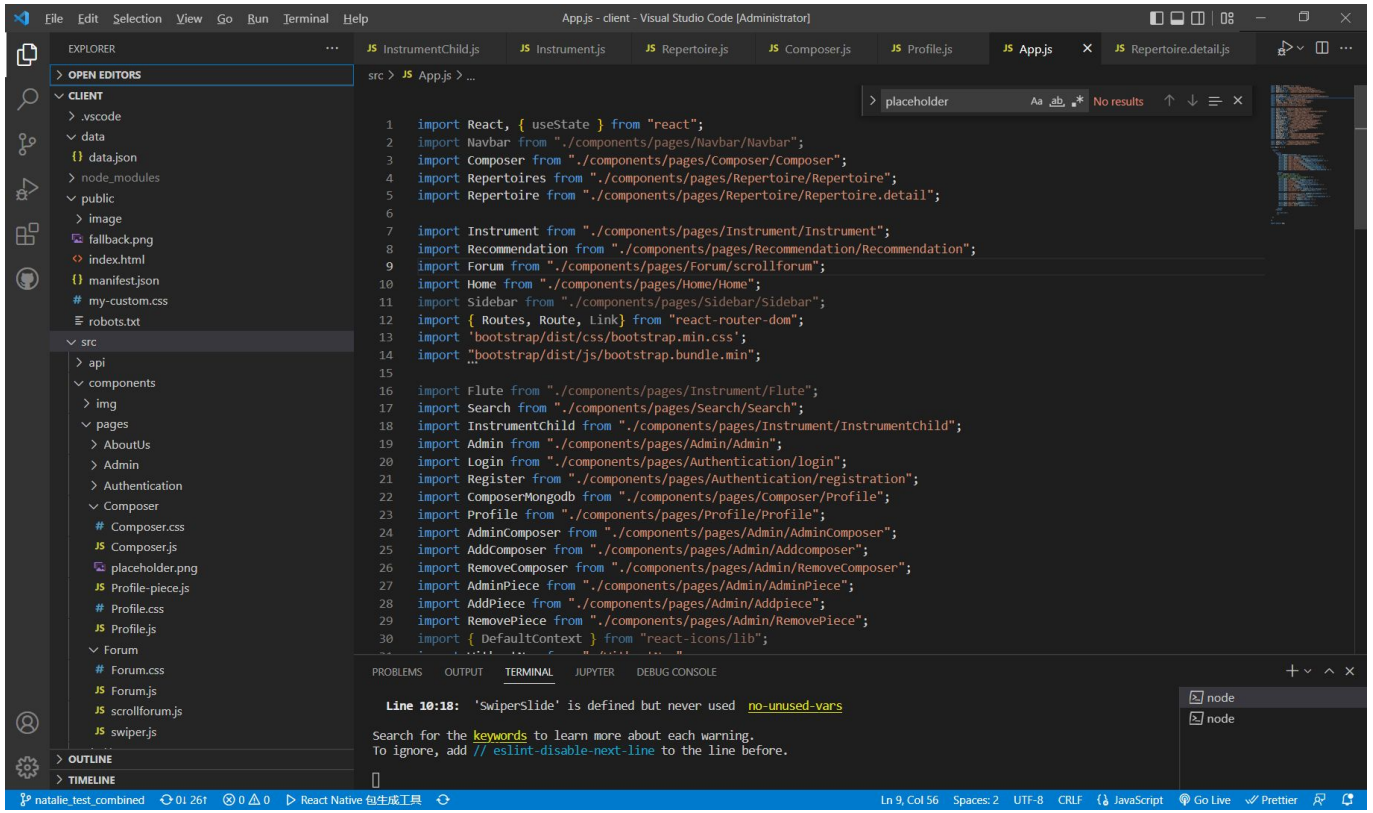
SWINBURNE UNIVERSITY OF TECHNOLOGY

Code Demonstration

Frontend

React JS

Bootstrap

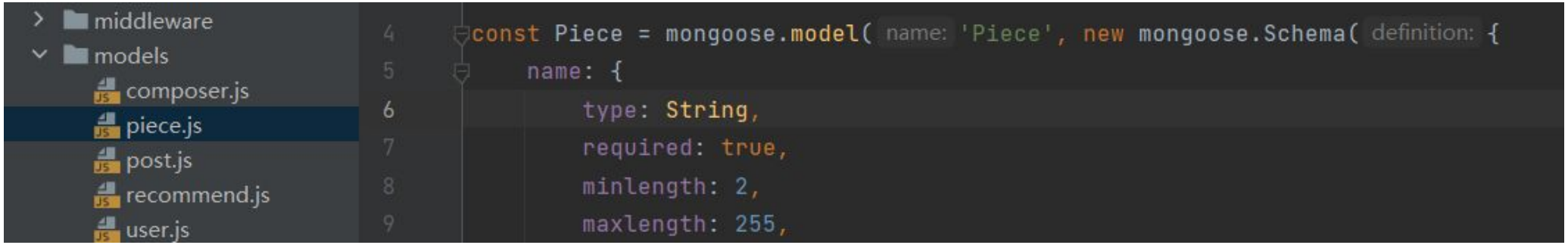


SWINBURNE UNIVERSITY OF TECHNOLOGY

Code Demonstration

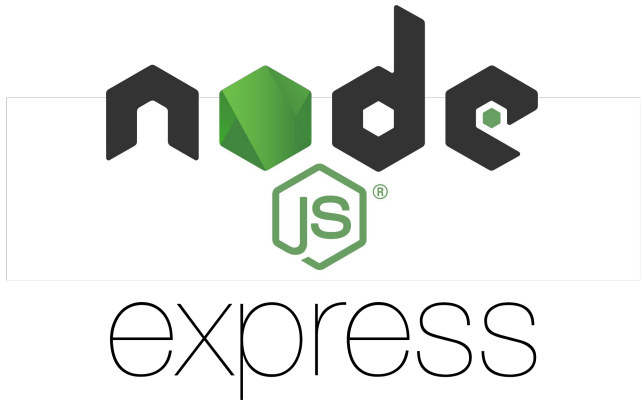
Database

Mongoose schema



Code Demonstration

Backend



routes
- admin.js
- auth.js
- comments.js
- composers.js
- pieces.js
- posts.js
- recommends.js
- search.js
- users.js

```
13  router.post( path: '/', auth, async (req :… , res :… ) => {
14      const { error } = validatePost(req.body);
15      if (error) return res.status( code: 400).send(error.details[0].message);
16      const user = await User.findById(req.user._id).select( arg: '-password');
17
18      const post = new Post( doc: {
19          title: req.body.title,
20          body: req.body.body,
21          user: {
```

Thank you !

Now let’s take a look at the website.