Using the Template

This document provides detailed guidance on using the CM2040 Database Networks and the Web mid-term coursework template.

Quick Start

Follow the instruc�ons in the README.md file to set up your computer and install and run the template.

Contents of the Template

The following table describes the contents of the template:

**Component**

public/

main.css

**Descrip�on**

The public folder contains sta�c files, such as css files and client-side javascript, that you want your web server to serve.

page1image30917024page1image30916816page1image30923056page1image30924928

|  |  |
| --- | --- |
| routes/  users.js | The routes folder contains your middleware route handlers.  Add files here to implement specific middleware func�onality that you want. In the template, a sample handler for user func�onality is provided. You may wish to develop that further if your applica�on requires user func�onality, or remove it if it doesn’t. |
| views/  add-user.ejs | The views folder contains your EJS templates, which are used to render your data-driven web pages.  Add addi�onal EJS files here and develop or delete the add- user.ejs file as required. |
| index.js | This is your main applica�on start point. It sets up Express connects to your database, loads all your route handlers and starts your web applica�on listening for HTTP requests.  Add calls to your route handers here. |

package.json

package-lock.json

.gitignore

These tell node.js what packages and versions of packages you are using as well as how to run your project. You shouldn’t need to edit these.

If you use git, this file will exclude files you don’t want to commit to git.

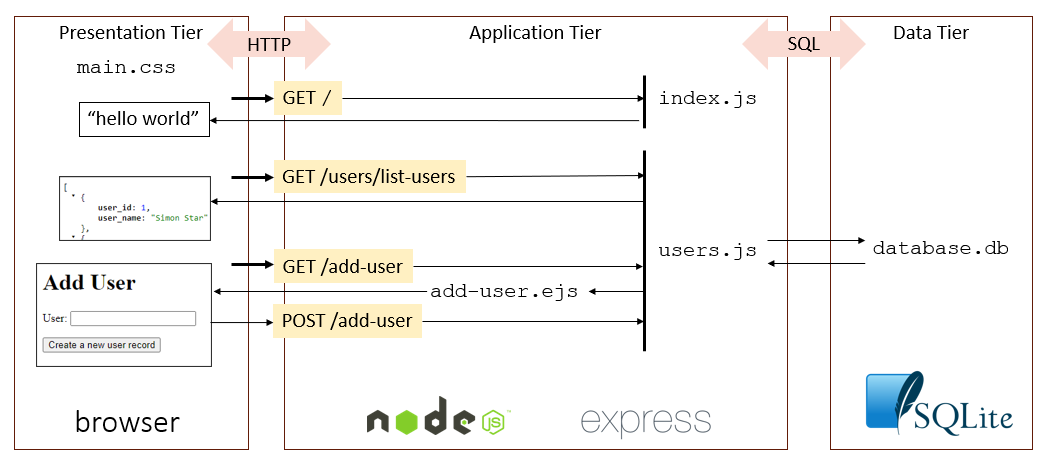
|  |  |
| --- | --- |
| db\_schema.sql | This defines your database structure. When you run npm run build-db or npm run build-db-win a new database is created (in a file called database.db) and the contents of this file are exectuted against the database.  Build your data model here. Changes made here won’t be applied un�l you exectute them against the sqlite database. |

|  |  |
| --- | --- |
| README.md | Contains instruc�ons for se�ng up your computer and running the sample code.  You should update this document so that it provides suitable instruc�ons for the applica�on that you develop. |

The Running Applica�on

The template as it stands par�ally implements a dynamic web applica�on that manages users and their email addresses.

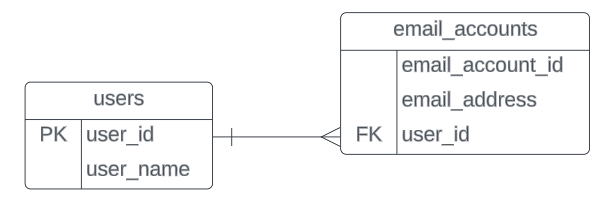
The following diagram shows how the interac�on between the layers of the 3-�er architecture are implemented in the template, together with the applica�on endpoints:



The Data Model

The following is the underlying data model, implemented in the file db\_schema.sql :

The following data is inserted into the two tables within the same db\_schema.sql file :



**users:**

**user\_id user\_name**

1. 1  Simon Star
2. 2  Dianne Dean
3. 3  Harry Hilbery

**email\_accounts:**

**email\_address user\_id**

simon@gmail.com 1 simon@hotmail.com 1 dianne@yahoo.co.uk 2

page2image31340144page2image31340352page2image31340560page2image31340768page2image31340976page2image31341184page2image31341392page2image31341600page2image13969248page2image13966336

Working with SQLite

SQLite works a litle differently to MySQL but all of the key concepts are the same. You can find the documenta�on for the sqlite3 node module here: htps://github.com/TryGhost/node-sqlite3/wiki/API  
You can find tutorials on using the sqlite3 node module here: htps://www.sqlitetutorial.net/sqlite-nodejs/

For a general tutorial on using SQLite start here:

htps://www.sqlitetutorial.net/

The SQLite Command Line

When you install SQLite, you will have a command line tool installed. You can use this to run SQL commands on your database so you can check the contents of tables, check the structure of the database and prac�ce your SQL queries before embedding them in your node.js code.

Start the command line tool by typing sqlite3 database.db from your command line or terminal, ensuring you are in the project directory containing your database:

Ensure you turn on the op�on to enforce foreign key constraints:

sqlite> **PRAGMA foreign\_keys=ON;**You can see a list of tables in the database by using the .tables command:

You can see a table defini�on by usin the.schema command:

You can also type in SQL statements, such as a SELECT command:

page3image30996496page3image30986512page3image30984432

c:\>**sqlite3 database.db**SQLite version 3.38.5 2022-05-06 15:25:27 Enter ".help" for usage hints.

page3image30982352

sqlite> **.tables** email\_accounts users

page3image30983184

sqlite> **.schema users** CREATE TABLE users (

user\_id INTEGER PRIMARY KEY AUTOINCREMENT,

user\_name TEXT NOT NULL

);

sqlite> **SELECT \* FROM users;** 1|Simon Star  
2|Dianne Dean  
3|Harry Hilbert

If you want to dump our the en�re database schema, including the INSERT commands used to insert data, you can use the .dump command:

|  |
| --- |
| sqlite> **.dump** |
| PRAGMA foreign\_keys=OFF;  BEGIN TRANSACTION;  CREATE TABLE users (  user\_id INTEGER PRIMARY KEY AUTOINCREMENT,  user\_name TEXT NOT NULL  );  INSERT INTO users VALUES(1,'Simon Star');  INSERT INTO users VALUES(2,'Dianne Dean');  INSERT INTO users VALUES(3,'Harry Hilbert’);  CREATE TABLE email\_accounts (  email\_account\_id INTEGER PRIMARY KEY AUTOINCREMENT, email\_address TEXT NOT NULL, user\_id INT, --the user that the email account belongs to FOREIGN KEY (user\_id) REFERENCES users(user\_id)  ); INSERT INTO email\_accounts VALUES(1,'simon@gmail.com',1); INSERT INTO email\_accounts VALUES(2,'simon@hotmail.com',1); INSERT INTO email\_accounts VALUES(3,'dianne@yahoo.co.uk',2); COMMIT; |

To exit the SQLite command line, use the .exit command: sqlite> .exit

Modifying the Schema

You will need to modify the database schema to implement your applica�on. You **must** do this by modifying db\_schema.sql. This allows us to review and recreate your database simply by running npm run build-db. Do NOT create or alter database tables through other means.

Suggested Next Steps

Spend some �me working with the template:

* Explore the file structure and code
* Read all the comments
* Try accessing each of the routes via the browser - make sure you understand what they do
* Try adding a new route to display all the email accounts for a par�cular user
* Try adding a new form to allow new email accounts to be added for a user

Once you are comfortable with the template you can start to plan your implementa�on of the mid- term project. You might start working on wireframes for your web pages and designing the data model before trying to write the node.js code.