**PERSONAL PORTFOLIO**

**BRENDAN ROTHWELL**

**N8540683**

**ARTEFACT LIST**

1. **PGAdminView.doc:** Screenshot of PGAdmin view of the PostgreSQL Database. I was the DBA of the group, and most of my work was done here. Designing, Creating and Maintaining the Database allowed the team to access the data they needed to create our app.
2. **DB Creation Script PostgreSQL Edited3.sql:** An example SQL script. The main script amongst several that were kept to act as a backup for our Database. This script was the main DB creation script amongst others that included small changes. This ongoing task provided our team some security throughout the project.
3. **MusicSchool DB ERD.pdf:** The ERD of the initial design of the Database. This allowed easy and large-scale creation of our database to be done quickly at the start of the project, allowing work to progress easily and avoiding any Database tasks being blockers for others in our team.
4. **Artefact4.doc:** I added gender fields to all registration forms. This allowed students to be able to be able to request a teacher of specific gender. See doc file for more details.
5. **Artefact5.doc**: I moved the common styling we already had into some common css pages to allow the css to work globally through our web app to reduce duplication of code and to allow faster styling of future pages. See doc file for more details.
6. **Researching JQuery and JQuery-UI Datepicker:** Since nobody in our group had any experience in using JQuery and our client group asked us to include a datepicker, I had to research how to use JQuery, and more specifically, the JQuery-UI datepicker. Unfortunately Mozilla Firefox does not accept the HTML5 ‘date’ input field type yet or this task would have been far simpler.

**See:**

<https://jqueryui.com/datepicker/> & <https://www.tutorialspoint.com/jqueryui/jqueryui_datepicker.htm>

1. **Artefact 7.doc:** I added a JQuery datepicker to any date fields in any forms. This provided some extra front-end validation for date input and also looked more appealing to the user. See doc file for more details.
2. **DBTables.doc:** This was a document I kept updated which outlined the database design/structure in an easy-to-understand format. This was used as an assistive guide for form creation and for back end work. Whenever a change was made to the DB, I would update this document immediately after.
3. **DatabaseChanges.doc:** This was a document in which Richard or Anneke would write down changes that were required and I would mark them off as red text once completed. This proved to be a good way of accurately referencing what was needed to be done and also showing that it had been done and when. This allowed us to reduce human error when working on names that had to be accurately spelled and named in an informative way. It also was a clear form of communicating table structure.
4. **CopyingDataScript – ExtraColumnValue.sql:** This was a script I wrote in order to copy data from an old table (which I moved to a backup schema name before creating a new schema) to it’s new equivalent table that had a new extra column. Then I had to add a numeric value for every row of the new column, only then being able to make the new column not-nullable. This is one of two data backup scripts I wrote, but was the more complicated of the two. This was done in addition to the DB and new table create scripts to serve as a fool-proof Database backup security system.
5. **Sprint 2 Powerpoint.pptx:** I prepared the initial draft of the presentation for the end of Sprint 2, to which other members such as Richard and Anneke were able to make some small adjustments. This allowed for faster preparation for the presentation and took some of the preparation workload and pressure off other team members.