

# CPSC 304 Project Cover Page

Milestone #: \_\_\_\_\_

Date: \_\_\_\_\_

Group Number: \_\_\_\_\_

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

## 2. Project Summary:

Our project is a centralized hub of information that allows users to access and contribute detailed information about the Pokémon video game series. Information about the game's Pokémon, trainers, gym leaders, moves, and more are to be tracked so users are able to develop strategies and keep track of the vast amount of information available in the Pokémon series. Users should be able to find information such as which trainers possess which Pokémon, what are the weaknesses or strengths of a type, or which Pokémon can learn certain moves.

## 3. Timeline and Task Breakdown:

- Each person is in charge of doing the GUI elements for their assigned queries.
- Each person will also be writing their backend API endpoints, notifications, data sanitation, and SQL queries for each of their assigned queries.
- Each person is also in charge of making examples for their sections and queries. This also includes explaining each query and adding their respective section to the SQL script for drop, recreate, and reload tables.
- Joshua and Matthew handle page routing and base look of the application.
- Joshua will put together the final version of the SQL script based on the work that all group members contribute for their sections.
- Matthew will handle the cover page, description of project, description of changes to schema design and schema turned in for Milestone 4.

We are splitting tasks based on queries so that everyone gets a chance to practice all parts of creating an application. This breakdown is going to allow us each to learn about each component of our application so we maximize our learning and can help each other out. Joshua and Matthew have the most experience with making projects so they will handle the page routing and setting up the base look of the application.

Each of the query breakdowns for each person are below.

Albert:

- (2.1.1) One relation to insert on
  - Insert onto the pokemon table and pokemon attributes tables
- (2.1.2) One relation to update
  - Update onto the pokemon table and pokemon attributes tables
- (2.1.3) One relation to delete
  - Delete relations in the pokemon table and pokemon attributes tables
- (2.1.4) One relation to selection on
  - Select pokemon based on types (can have multiple types)

Joshua:

- (2.1.6) One relation to join on

- Users should be able to see the gym leader's name, where they're located, and their pokemon roster
- (2.1.8) Aggregation with having
  - Users can use this information to learn which pokemon you can have in your party before they evolve
- (2.1.10) Division
  - Users should be able to find if a given list of pokemon are used in a trainer's roster. For example you want to find out if a trainer you will need to face will have some Pokemon that will counter yours.

Matthew:

- (2.1.5) One relation to project on
  - Users should be able to find the name of every single Pokemon in the game.
- (2.1.7) Aggregation with group by
  - Users should be able to count the number of pokemon by region to be able to figure out how many new pokemon they're able to find in a given region.
- (2.1.9) Nested aggregation with group by
  - Users should be able to find the Pokemon type with the strongest/weakest average power of their moves.

## TIMELINE:

Oct 25th:

- Ensure everybody has push/pull/branch access to the project repo
- Ensure that everyone contributes at least one commit
- Previous milestones must be added to the project repo
- Iron out tasks and work distribution, and everyone must agree with the plan
- Book a meeting slot with the TA to discuss milestone 3
- Explanations/mockups for SQL queries are complete

Nov 1st

- Set up next/react skeletons and run the default example app
- Set up AWS postgresSQL instance and connect with backend
- Assigned SQL queries are drafted up and mostly syntactically/semantically correct
- Explanations and documentation for SQL queries are complete

Nov 8th

- SQL reset/loading script are finalized and complete
- Basic page navigation and layout for frontend is implemented
- Basic page routing on local host is complete
- Backend API endpoints are defined and are ready to be implemented

Nov 15th

- ~50% Backend API endpoints are implemented with notifications, data sanitation and error handling built in
- Corresponding frontend API endpoints are usable, not pretty

Nov 22nd

- ~90-100% Backend API endpoints are implemented
- Corresponding frontend API endpoints usable and pretty
- Cover page, project description and security are implemented
- Endpoints written by others are tested thoroughly
- Project schema and changes noted in project description

DUE Nov 29th

- Project is perfectly usable as outlined in project description
- Everyone should read over and have commented on the project cover page, description, schema description and changes.