# Epics

**Front End:**

* HTML Webpage (story): As a dev, I want a webpage, so that users can interact with my application.
  + Task: Create a webpage in HTML
  + Given the user navigates executed the jar file, when they navigate to the correct URL, then the webpage should load.
* JavaScript Functions (story): As a dev, I want to add functionality to my webpage, so that it can accommodate my backend functions.
  + Task: Add Fetch API methods.
  + Given the user is in the Webpage, when they try to interact with the inputs and buttons, then they should be able to add items to the database.
* CSS Styling (story): As a dev, I want to customise my webpage, so that it looks presentable.
  + Task: Add CSS/Bootstrap styling.
  + Given the user is in the Webpage, when they look at it, then they should be able see elements presented in a readable manner.

**Backend:**

* H2 Database (story): As a dev, I want a database, so that I can store information.
  + Task: Link a H2 database.
  + Given the user on the webpage, when they try to add items, then they should stored to the database.
* Java Classes (story): As a dev, I want add classes and objects to my java application, so that it can function.
  + Task: Add domain, service and controller classes for songs.
  + Task: Add domain, service and controller classes for artists.
  + Given the application is started, when it boots up, then it should run properly.
* Songs CRUD (story): As a dev, I want to add CRUD functions for songs, so that songs can be added and modified within my application.
  + Task: Add CRUD functions for songs.
  + Given the application is running, when the methods are called, they should correctly execute CRUD functions.
* Artists CRUD (story): As a dev, I want to add CRUD functions for artists, so that artists can be added and modified within my application.
  + Task: Add CRUD functions for artists.
  + Given the application is running, when the methods are called, they should correctly execute CRUD functions.
* Pom (story): As a dev, I want to add dependencies to my Pom file, so that I can use features like testing.
  + Task: Add dependencies.
  + Given that the application is written, when plugins needing dependencies are used, there should not be any errors.

**Testing:**

* Selenium (story): As a dev, I want to add tests, so that make sure my features work.
  + Task: Write tests for the Frontend
  + Given the tests have been written, when they are run, then they should show either success or failure.
* Mockito/JUnit tests (story): As a dev, I want to add tests, so that make sure my features work.
  + Task: Write tests for the Backend.
  + Given the tests have been written, when they are run, then they should show either success or failure.

**README**:

* README Instructions (story)**:** As a Developer, I want to create a README, so that users know how to use the application.
  + Task: Edit README file.
  + Given the instructions have been written, when the user reads them, then they should understand how to run the application.