Build the Back End for an e-Commerce Site

* Create a back end for an e-commerce website using latest technologies
* Given is a functional Express.js API
* When the user adds:
  + Database name,
  + MySQL username,
  + MySQL password to an environment variable file
    - Then user can connect to a database using ***Sequelize***
* When the user enters *schema* and *seed* commands:
  + a development database is created and is seeded with test data
* When the user enters the command to invoke the application
  + The server is started and the Sequelize models are synced to the MySQL database
* When the user opens the API GET routes in Insomnia for categories, products, or tags
  + the data for each of these routes is displayed in a formatted JSON
* When the user tests the API POST, PUT, and DELETE routes in ***Insomnia***
  + Test results should successfully create, update, and delete data in my database

**Getting Started**

Technologies needed:

* **MySQL2** – connects Express.js API to MySQL database
* **Sequelize** – connects Express.js API to MySQL database
* **Express.js** – connects Express.js API to MySQL database
* **Dotenv** – use for environment variables to store sensitive data

Use ***schema.sql*** (in db folder) to create your database using MySQL shell commands.

Use **dotenv** variables to store sensitive data.

**Database Models**

Required models and requirements for each model:

* Category
  + id
  + Integer
  + Doesn't allow null values
  + Set as primary key
  + Uses auto increment
  + category\_name
  + String
  + Doesn't allow null values
* Product
  + id
  + Integer
  + Doesn't allow null values
  + Set as primary key
  + Uses auto increment
  + product\_name
  + String
  + Doesn't allow null values
  + price
  + Decimal
  + Doesn't allow null values
  + Validates that the value is a decimal
  + stock
  + Integer
  + Doesn't allow null values
  + Set a default value of 10
  + Validates that the value is numeric
  + category\_id
  + Integer
  + References the category model's id
* Tag
  + id
  + Integer
  + Doesn't allow null values
  + Set as primary key
  + Uses auto increment
  + tag\_name
  + String
* ProductTag
  + id
  + Integer
  + Doesn't allow null values
  + Set as primary key
  + Uses auto increment
  + product\_id
  + Integer
  + References the product model's id
  + tag\_id
  + Integer
  + References the tag model's id