**User Story**

AS A manager at an internet retail company

I WANT a back end for my e-commerce website that uses the latest technologies

SO THAT my company can compete with other e-commerce companies

**Acceptance Criteria**

GIVEN a functional Express.js API

WHEN I add my database name, MySQL username, and MySQL password to an environment variable file

THEN I am able to connect to a database using Sequelize

WHEN I enter schema and seed commands

THEN a development database is created and is seeded with test data

WHEN I enter the command to invoke the application

THEN my server is started and the Sequelize models are synced to the MySQL database

WHEN I open API GET routes in Insomnia Core for categories, products, or tags

THEN the data for each of these routes is displayed in a formatted JSON

WHEN I test API POST, PUT, and DELETE routes in Insomnia Core

THEN I am able to successfully create, update, and delete data in my database

### Database Models

Your database should contain the following four models, including the requirements listed 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

### Associations

You'll need to execute association methods on your Sequelize models to create the following relationships between them:

* Product belongs to Category, as a category can have multiple products but a product can only belong to one category.
* Category has many Product models.
* Product belongs to many Tag models. Using the ProductTag through model, allow products to have multiple tags and tags to have many products.
* Tag belongs to many Product models.