\*INTRODUCTION\* Internet retail, also known as e-commerce, is the largest sector of the electronics industry, having generated an estimated US$29 trillion in 2017 (Source: United Nations Conference on Trade and Development). E-commerce platforms like Shopify and WooCommerce provide a suite of services to businesses of all sizes. Due to the prevalence of these platforms, developers should understand the fundamental architecture of e-commerce sites. Your challenge is to build the back end for an e-commerce site. You’ll take a working Express.js API and configure it to use Sequelize to interact with a MySQL database. \*STARTER CODE \* PLEASE USE INITIAL CODING FILE (ATTACHED) AS THE BASIS FOR THIS ORDER!!! \*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 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 THEN I am able to successfully create, update, and delete data in my database \*MOCK-UP SCREENSHOTS\* Attached! Please utilize as a (DEMO SAMPLE)! \*ADDITIONAL INFORMATION\* Getting Started You’ll need to use the MySQL2Links to an external site. and SequelizeLinks to an external site. packages to connect your Express.js API to a MySQL database and the dotenv packageLinks to an external site. to use environment variables to store sensitive data, like your MySQL username, password, and database name. Use the schema.sql file in the db folder to create your database using MySQL shell commands.

Use environment variables to store sensitive data, like your MySQL username, password, and database name. 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. Fill Out the API Routes to Perform RESTful CRUD Operations Fill out the unfinished routes in product-routes.js, tag-routes.js, and category-routes.js to perform create, read, update, and delete operations using your Sequelize models. Seed the Database After creating the models and routes, run npm run seed to seed data to your database so that you can test your routes. Sync Sequelize to the Database on Server Start Create the code needed in server.js to sync the Sequelize models to the MySQL database on server start.