MySQL

- Create movie database and create the schemas
- Populate the database with some pre-made data using an SQL script

Maven project creation and setup

- Create a new maven project and select “Create a simple project”
- Fill in the groupId and artifactId
- Modify the pom.xml to add necessary dependencies (JPA, Spring security, MySQL database driver, swagger, swagger UI, testing frameworks, spring security test, thymeleaf, bootstrap, jquery)
- Add the necessary maven plugins for spring boot
- Create an application.yml file and specify the database credentials

Bootstrapping our web app

- Create a MovieApplication class with a main method
- Create a SecurityConfiguration class that specifies authorized usernames and passwords
- Create the Java entity models that represent tables in the database
- Create repository interfaces for each model that extends JpaRepository
- Create DTOs for each model
- Create service classes for each entity that perform CRUD operations on the repositories
- Create controller classes for each entity
- The controller classes specify REST api mappings to code that invokes the service layers to perform the requested CRUD operation
- Create a SessionController class to manage the order and checkout experience
- This controller uses an in-memory map to keep track of the checkout details
- It uses cookies to identify and keep track of the user
- Add CORS support by creating a CorsConfiguraiton class that implements WebMvcConfigurer
- This class defines a cors mapping to allow all cross origins requests