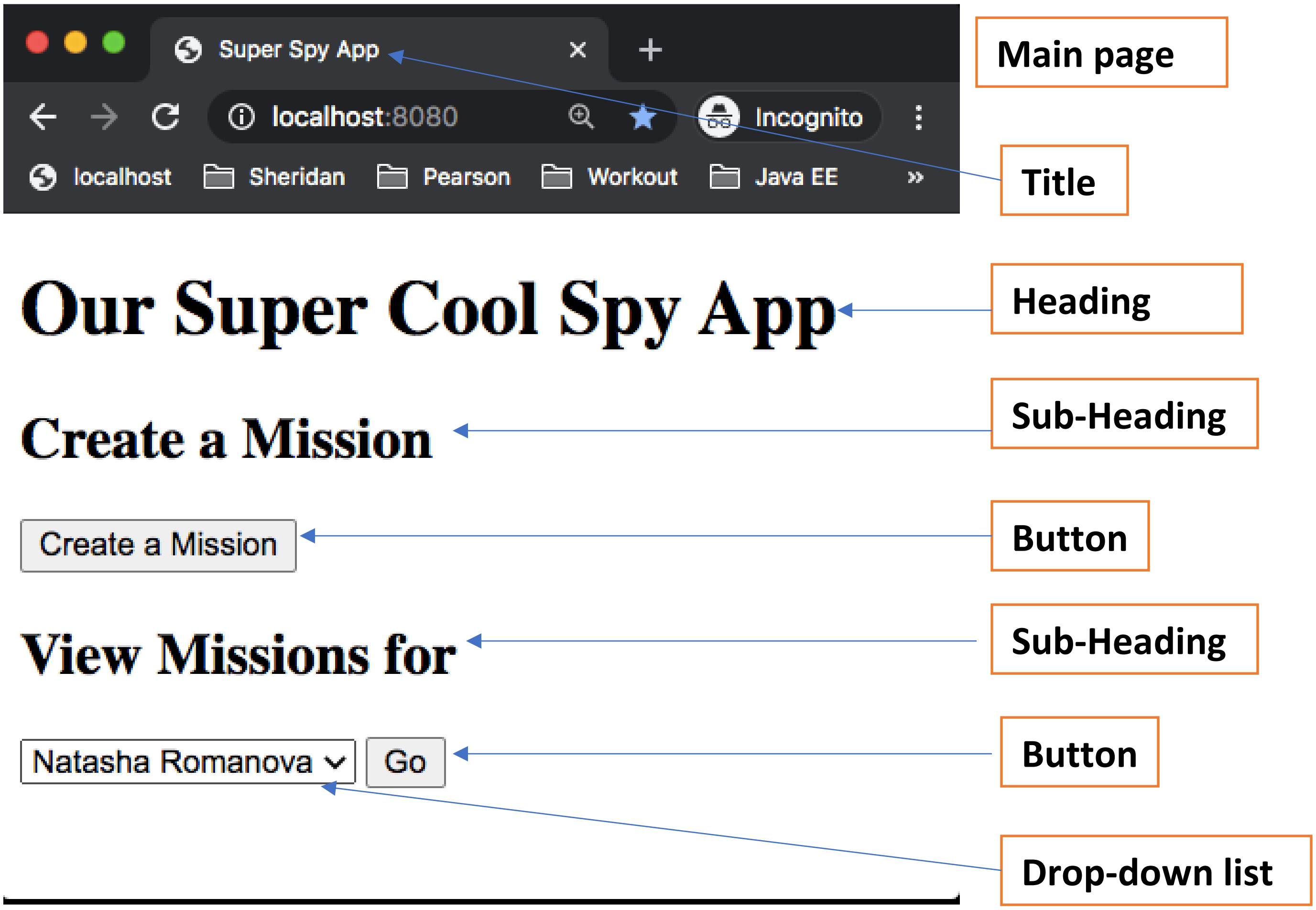
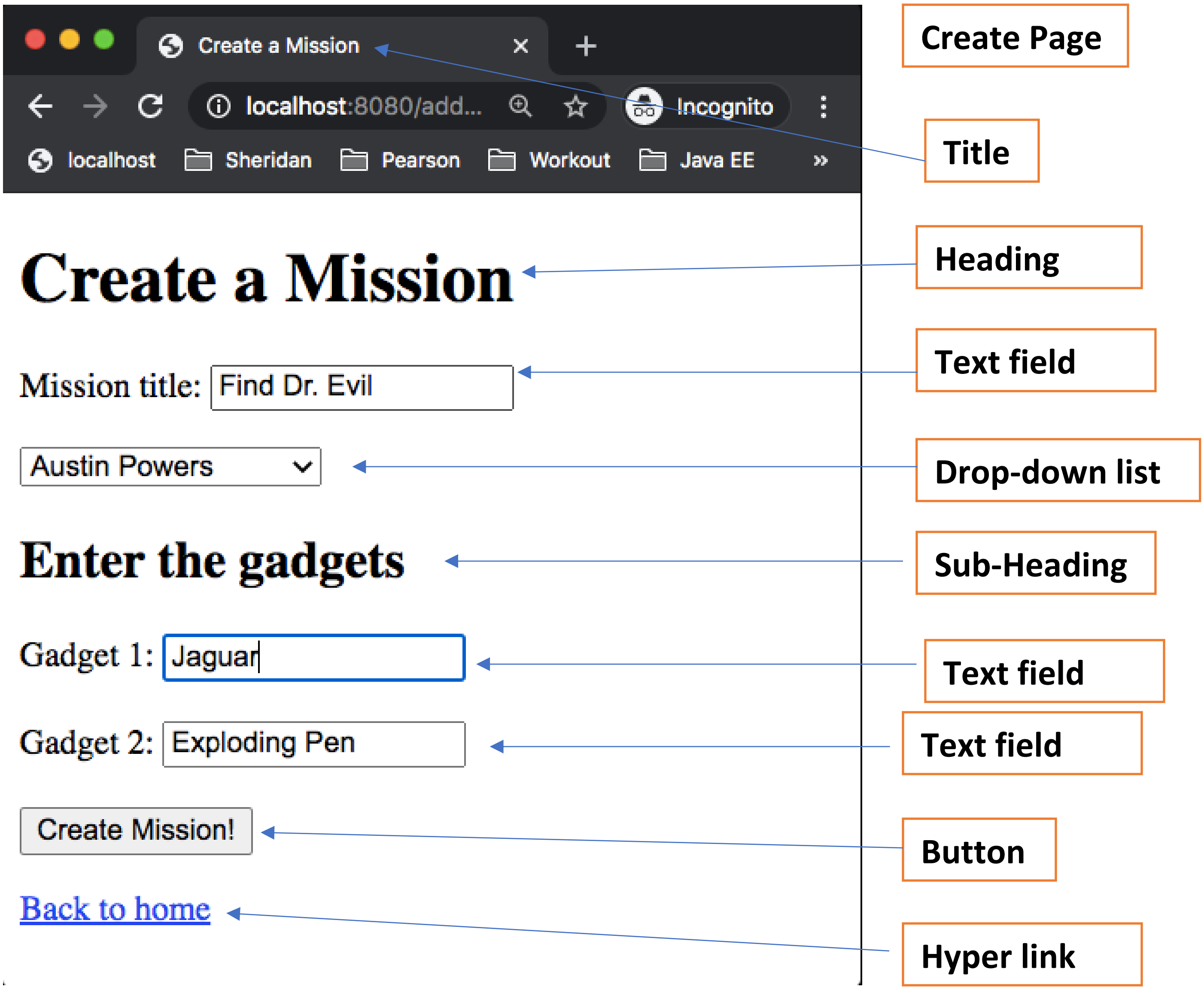
PROG32758 - Assignment 2

This assignment is to be done individually and is subject to the school Policy and Guidelines on Academic Honesty.

In this assignment you will create a simple web application.



- Create a Mission button loads the Create page.
- Drop down list menu should update after a new mission gets created.



Create a Mission

localhost:8080/add...

Create a Mission

Mission title: Find Dr. Evil

Austin Powers

Enter the gadgets

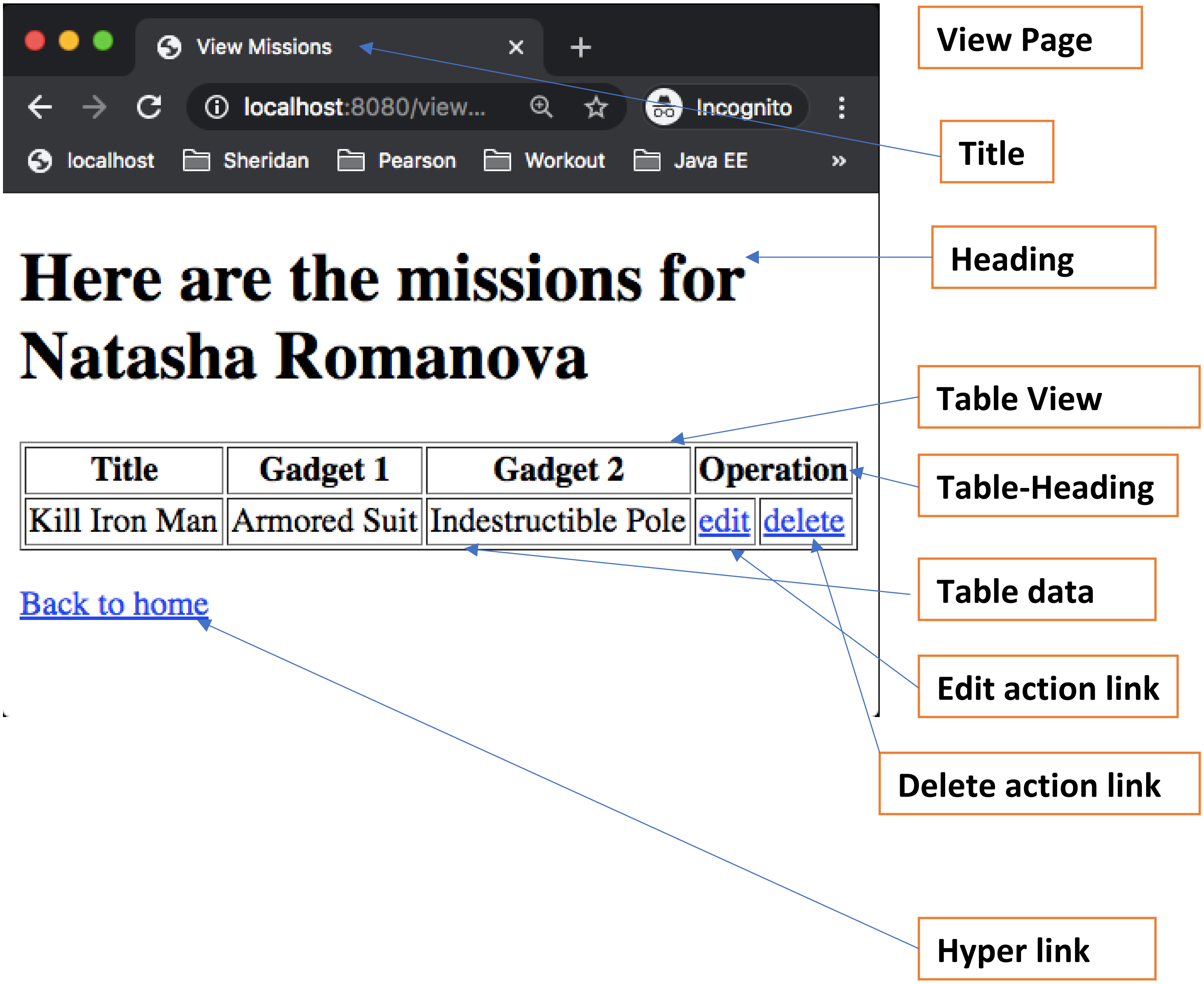
Gadget 1: Jaguar

Gadget 2: Exploding Pen

Create Mission!

Back to home

- Drop down list menu shows all the comic character’s names.
- After clicking on the Create Mission! button your application should go back to the main page.

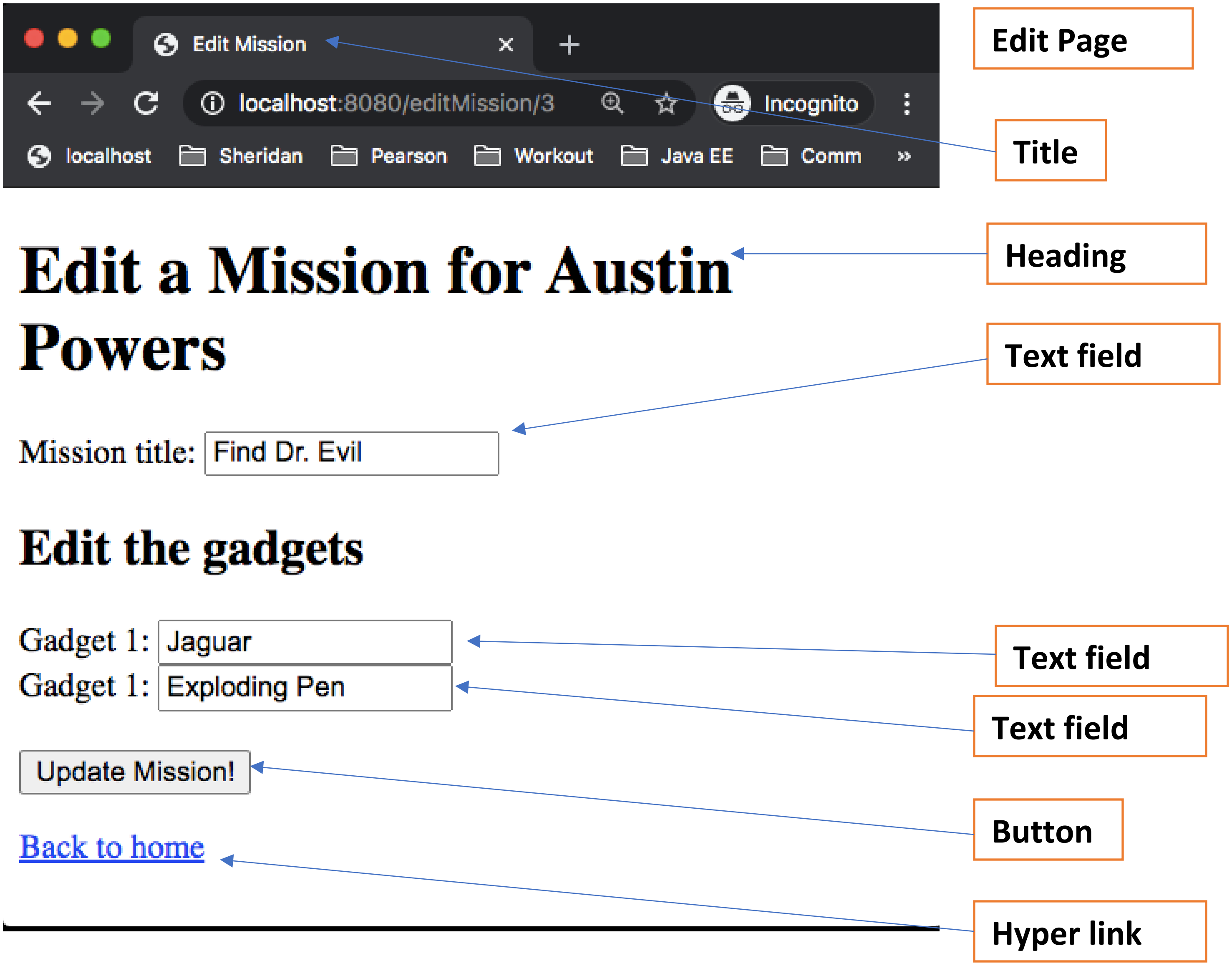


Here are the missions for Natasha Romanova

Title	Gadget 1	Gadget 2	Operation	
Kill Iron Man	Armored Suit	Indestructible Pole	edit	delete

Back to home

- View page loads the data about a particular comic character mission information which was chosen on the main page
- Data should load in the table format and if a particular character has multiple mission's then edit and delete actions should be shown for each row of mission.
- "edit" action link should load the Edit page.
- "delete" action link should delete that particular link and loads the page again with left over mission to show but if there is no data/mission left to show for that particular character then loads the main page.



Edit a Mission for Austin Powers

Mission title: Find Dr. Evil

Edit the gadgets

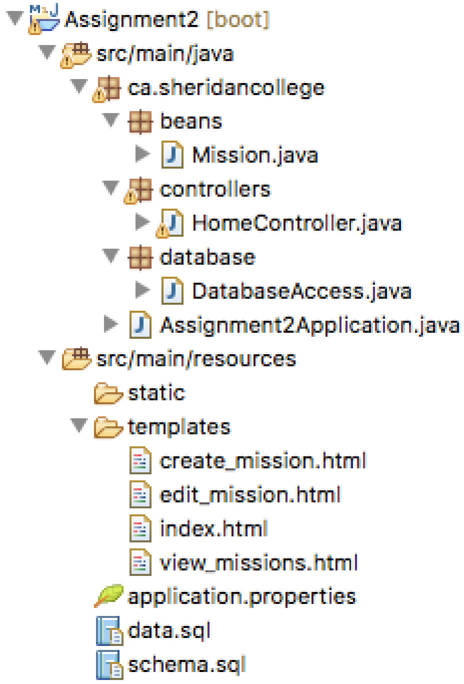
Gadget 1: Jaguar
Gadget 1: Exploding Pen

- “Update Mission!” should update the particular record and loads the main page after editing.

- You should create a Spring Boot application with Spring Web, Thymeleaf, Lombok, H2 Database and JDBC SQL dependencies.
- You must use the following;
  - Best practices as taught in class for database access (using H2):
    - NamedParameterJdbcTemplate

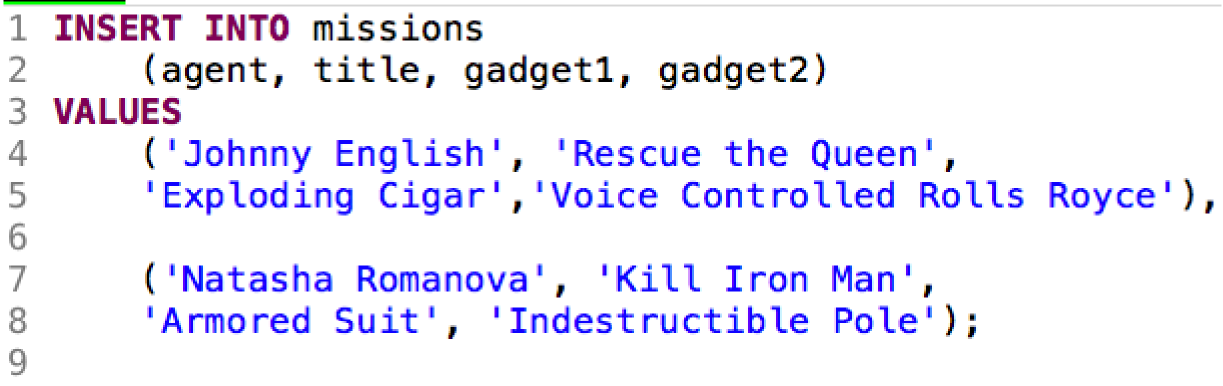
        - MapSqlParameterSource
        - BeanPropertyRowMapper
    - Thymeleaf in all dynamic pages and *Form Binding* in Create/Edit mission html pages.
    - *Path Variables* for the edit/delete functionality.
- You are more than welcome to make it look better than my "programmer pages," but make sure you have all the back-end stuff working first and don't make too many changes to the functionality.
- ***You may modify the application to fit something that interests you more, but make sure it has the same functionality, input types and one-to-many relationship as the Spy App. If you're not sure, just ask!***
- Make sure you look at the rubric on slate before you final submission!

Your project should have a structure very similar to the following.



```
▼ Assignment2 [boot]
  ▼ src/main/java
    ▼ ca.sheridancollege
      ▼ beans
        ▶ Mission.java
      ▼ controllers
        ▶ HomeController.java
      ▼ database
        ▶ DatabaseAccess.java
      ▶ Assignment2Application.java
  ▼ src/main/resources
      static
    ▼ templates
        create_mission.html
        edit_mission.html
        index.html
        view_missions.html
      application.properties
      data.sql
      schema.sql
```

data.sql

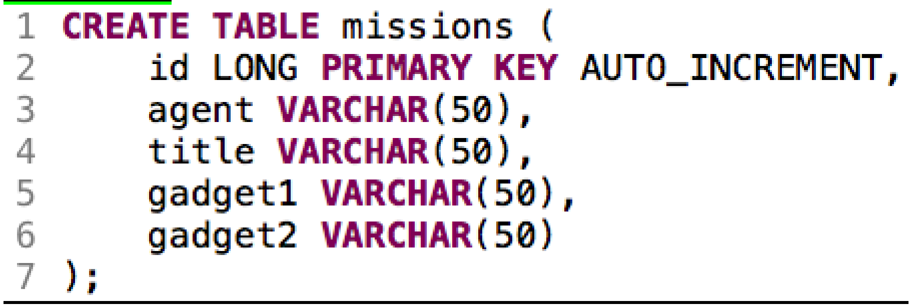


```sql
INSERT INTO missions
     (agent, title, gadget1, gadget2)
VALUES
     ('Johnny English', 'Rescue the Queen',
     'Exploding Cigar','Voice Controlled Rolls Royce'),

     ('Natasha Romanova', 'Kill Iron Man',
     'Armored Suit', 'Indestructible Pole');

```

schema.sql



```sql
CREATE TABLE missions (
     id LONG PRIMARY KEY AUTO_INCREMENT,
     agent VARCHAR(50),
     title VARCHAR(50),
     gadget1 VARCHAR(50),
     gadget2 VARCHAR(50)
);
```

Submission Criterion

- Submit your zipped project (not rar or 7zip)
- Make sure to look at the marking scheme on the next page

Marking Scheme

Part 1: Adding data -Records can be added to the database properly -Proper use of mission bean -Add page created properly using appropriate CSS and selection options. -All required information is collected properly -Links to the root page	/25
Part 2: Dummy Records and overall working -2 missions as provided for the data.schema into the database. (you can update data.schema to load more missions as well) -This can be done through a hyperlink/button on the root page or in the sql file.	/20
Part 3: View All Records -HTML page that displays particular mission information into a table. -Makes use of appropriate CSS. -Links to the root page	/25
Part 4: Edit and Delete -Edit (15 marks) and Delete (15 marks)	/30
Deductions (These are deductions on any marks received above): -Program does not make use of proper structure -Program does not use appropriate Spring Boot or Thymeleaf libraries -Program does not link to other pages appropriately -SQL setup file not included, or not properly constructed. -Late	
Total	/100