**Exercise 9: Creating a Spring Boot Application**

// Book.java

@Entity

public class Book {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String title;

private String author;

// Getters and setters

}

// BookRepository.java

public interface BookRepository extends JpaRepository<Book, Long> {

}

// BookController.java

@RestController

@RequestMapping("/api/books")

public class BookController {

@Autowired

private BookRepository bookRepository;

@GetMapping

public List<Book> getAllBooks() {

return bookRepository.findAll();

}

@GetMapping("/{id}")

public Book getBook(@PathVariable Long id) {

return bookRepository.findById(id).orElseThrow();

}

@PostMapping

public Book createBook(@RequestBody Book book) {

return bookRepository.save(book);

}

@PutMapping("/{id}")

public Book updateBook(@PathVariable Long id, @RequestBody Book book) {

Book existingBook = bookRepository.findById(id).orElseThrow();

existingBook.setTitle(book.getTitle());

existingBook.setAuthor(book.getAuthor());

return bookRepository.save(existingBook);

}

@DeleteMapping("/{id}")

public void deleteBook(@PathVariable Long id) {

bookRepository.deleteById(id);

}

}

// application.properties

spring.datasource.url=jdbc:h2:mem:library

spring.datasource.username=sa

spring.datasource.password=

spring.jpa.hibernate.ddl-auto=create-drop

// LibraryManagementApplication.java

@SpringBootApplication

public class LibraryManagementApplication {

public static void main(String[] args) {

SpringApplication.run(LibraryManagementApplication.class, args);

}

}