

Seller

Provider entity class

```
package csc340.BookbridgeBackend;

import jakarta.persistence.*;
import java.util.List;

@Entity 20 usages
@Table(name = "providers")
public class Provider extends User {

    private String businessName; 3 usages

    @OneToMany(mappedBy = "provider") 3 usages
    private List<Book> books;

    public Provider() { super(name: "", email: "", accountStatus: "", role: "Provider"); // Set default role as Provider }

    public Provider(String name, String email, String accountStatus, String businessName, List<Book> books) { no usages
        super(name, email, accountStatus, role: "Provider"); // Set role as Provider
        this.businessName = businessName;
        this.books = books;
    }

    public String getBusinessName() { return businessName; }

    public void setBusinessName(String businessName) { this.businessName = businessName; }

    public List<Book> getBooks() { return books; }

    public void setBooks(List<Book> books) { this.books = books; }
}
```

Provider controller class (part 1, 2 and 3)

```

package csc340.BookbridgeBackend;

import ...

@Controller no usages
@RequestMapping("/providers")
public class ProviderController {

    @Autowired 9 usages
    private ProviderService providerService;

    @GetMapping("/all") no usages
    public String getAllProviders(Model model) {
        model.addAttribute(attributeName: "providers", providerService.getAllProviders());
        return "providers"; // Maps to "providers.html"
    }

    @GetMapping("/{id}") no usages
    public String getProviderById(@PathVariable Long id, Model model) {
        model.addAttribute(attributeName: "provider", providerService.getProviderById(id)
            .orElseThrow(() -> new RuntimeException("Provider not found with ID: " + id)));
        return "provider-details"; // Maps to "provider-details.html"
    }

    @GetMapping("/new") no usages
    public String showCreateProviderForm(Model model) {
        model.addAttribute(attributeName: "provider", new Provider());
        return "providers-form"; // Maps to "provider-form.html"
    }

    @PostMapping("/new") no usages
    public String createProvider(@ModelAttribute Provider provider) {
        providerService.createProvider(provider);
        return "redirect:/providers/all";
    }
}

```

```

@GetMapping("/update/{id}") no usages
public String showUpdateProviderForm(@PathVariable Long id, Model model) {
    model.addAttribute(attributeName: "provider", providerService.getProviderById(id)
        .orElseThrow(() -> new RuntimeException("Provider not found with ID: " + id)));
    return "provider-form";
}

@PostMapping("/update/{id}") no usages
public String updateProvider(@PathVariable Long id, @ModelAttribute Provider provider) {
    providerService.updateProvider(id, provider);
    return "redirect:/providers/all";
}

@GetMapping("/delete/{id}") no usages
public String deleteProvider(@PathVariable Long id) {
    providerService.deleteProvider(id); // Call the service method
    return "redirect:/providers/all";
}

// View books listed by a provider
@GetMapping("/{id}/books") no usages
public String viewProviderBooks(@PathVariable Long id, Model model) {
    model.addAttribute(attributeName: "books", providerService.getBooksByProvider(id));
    return "provider-books"; // Maps to "provider-books.html"
}

// Create a new book listing for a provider
@GetMapping("/{id}/books/new") no usages
public String showCreateBookForm(@PathVariable Long id, Model model) {
    model.addAttribute(attributeName: "book", new Book());
    model.addAttribute(attributeName: "providerId", id);
    return "book-form"; // Maps to "book-form.html"
}

```

```

package csc340.BookbridgeBackend;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

@Repository 2 usages
public interface ProviderRepository extends JpaRepository<Provider, Long> {}

```

Provider Repository class

```

// Get provider by ID
public Optional<Provider> getProviderById(Long providerId) { return providerRepository.findById(providerId); }

// Get books listed by a specific provider
public List<Book> getBooksByProvider(Long providerId) { 1 usage
    return bookRepository.findByProviderId(providerId); // Assuming this method exists in BookRepository
}

// Create a book posting
@ public Book createBookForProvider(Long providerId, Book book) { 1 usage
    Provider provider = providerRepository.findById(providerId)
        .orElseThrow(() -> new ResourceNotFoundException("Provider not found with id " + providerId));
    book.setProvider(provider); // Link book to provider
    return bookRepository.save(book);
}

// Delete provider
public void deleteProvider(Long providerId) { 1 usage
    providerRepository.findById(providerId)
        .orElseThrow(() -> new RuntimeException("Provider not found with ID: " + providerId));
    providerRepository.deleteBy(providerId);
}

private static class ResourceNotFoundException extends RuntimeException { 1 usage
    public ResourceNotFoundException(String message) { super(message); }
}

// Get all orders for a specific provider
public List<Order> getAllOrdersForProvider(Long providerId) { 1 usage
    return orderRepository.findAllOrdersByProviderId(providerId);
}

```

Provider Service class

As the provider/seller, I can begin to add books to the site by pressing the “add new book”

| Books | | | | |
|--------------|--------|-------|-------------|--|
| Add New Book | | | | |
| Title | Author | Genre | Seller Name | Actions |
| 999 | 999 | 999 | 99 | View Edit Delete |

As the provider/seller, I am adding a book that I want to post to the site and I can press save so that it can be on the site

Add New Book

Title:
 Author:
 Genre:
 Seller Name:

| Books | | | | |
|------------------------------|----------|------------|--------------|--|
| Add New Book | | | | |
| Title | Author | Genre | Seller Name | Actions |
| ggg | ggg | ggg | gg | View Edit Delete |
| computer science 340 | Joy Sean | non-ficton | Joseph Grier | View Edit Delete |

We can now see the book now being added to the site

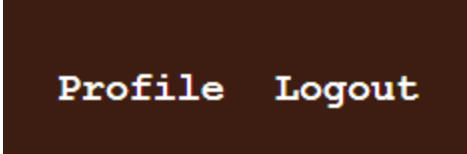
| | | | | | | | | | |
|--------------------------|----------------------|----------------------|------------------------|---|----------|------------|--------------|----------------------|------|
| <input type="checkbox"/> | Edit | Copy | Delete | 8 | Joy Sean | non-ficton | Joseph Grier | computer science 340 | NULL |
|--------------------------|----------------------|----------------------|------------------------|---|----------|------------|--------------|----------------------|------|

We can now see the book in uploaded on to the database

| | | |
|----------------------|----------------------|------------------------|
| View | Edit | Delete |
|----------------------|----------------------|------------------------|

I can also view, edit and delete any books on the site that I have uploaded

Buyer



Profile Logout

The buyer is able to look into their profile and can login and logout when they want to from the site



Search for books... Search

The buyer can search any book they would like to to further there education

College Courses

[CSC 101: Introduction
to Computer Science](#)

[MAT 201: Calculus I](#)

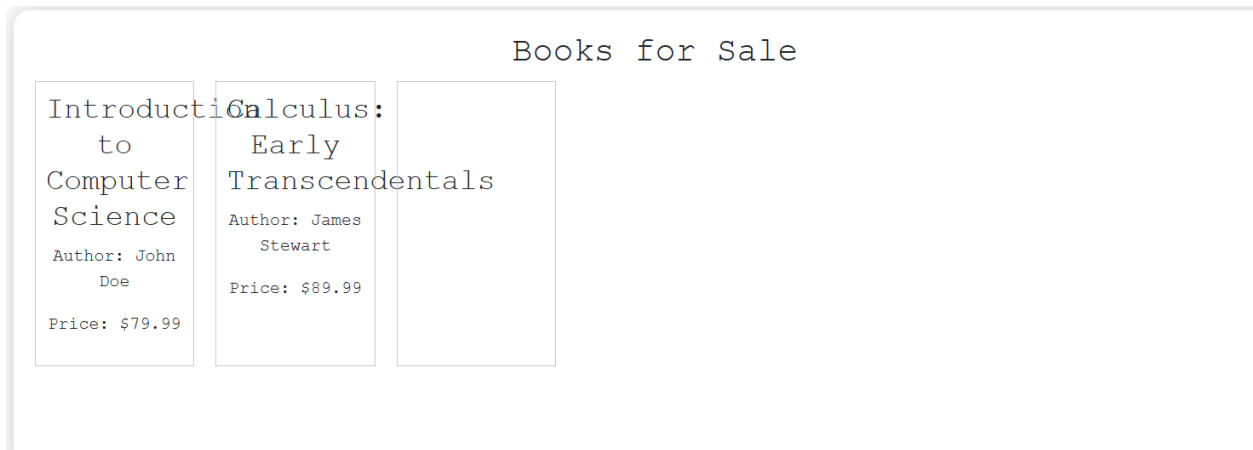
[ENG 202: English
Literature](#)

[HIS 301: World
History](#)

[BIO 102: Biology
Basics](#)

[CHE 110: General
Chemistry](#)

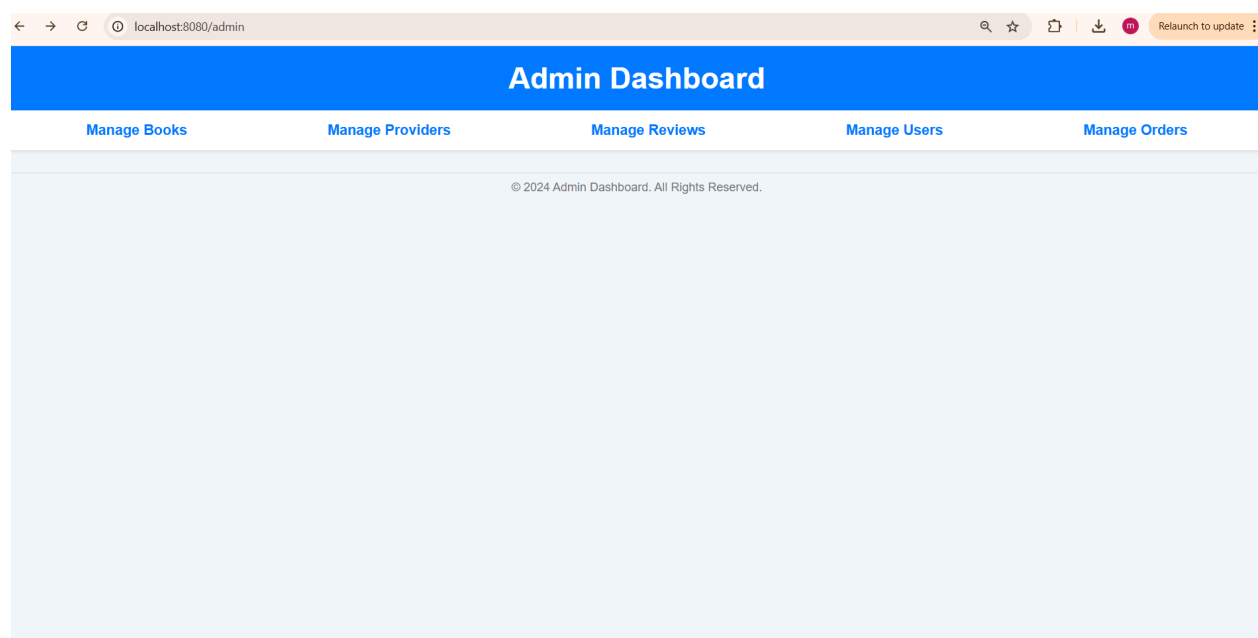
**The user will be able to see the list of book courses
available**



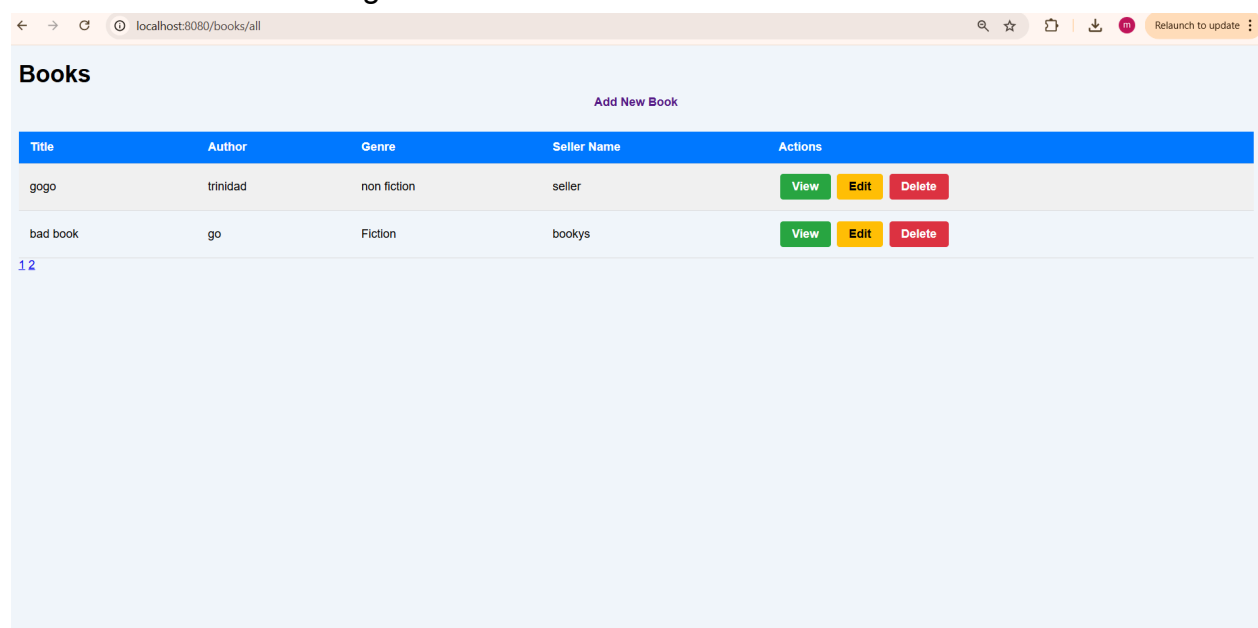
The user will also be able to see if there are any books for sale at the moment

SysAdmin

This is the main page for admin, shows all the options



This is where I can manage the books



This is where I can manage the providers

localhost:8080/providers/all

Relaunch to update

Providers

Add New Provider

| Name | Email | Business Name | Status | Actions |
|-----------------|-----------------------------|-----------------------|--------|----------------------------|
| updatedProvider | updatedProvider@example.com | Updated Awesome Books | active | <div>BooksEditDelete</div> |
| Provider Name | provider@example.com | Provider Business | active | <div>BooksEditDelete</div> |
| providernext | providernext@example.com | great Books | active | <div>BooksEditDelete</div> |
| provider123 | provider123@example.com | Awesome Books | active | <div>BooksEditDelete</div> |
| goodbooks | alice@example.com | goodbooks | ACTIVE | <div>BooksEditDelete</div> |

1

2

This is where I can manage the reviews

localhost:8080/reviews/all

Relaunch to update

Reviews

Create New Review

| Customer | Book Title | Rating | Actions |
|----------|------------|--------|---------|
|----------|------------|--------|---------|

1

2

This is where I can manage users

Users

Add New User

| Name | Email | Role | Account Status | Actions |
|---------------------|-----------------------------|----------|----------------|---|
| UpdatedCustomerName | customer@example.com | Customer | banned | <button>Edit</button> <button>Delete</button> |
| updatedProvider | updatedProvider@example.com | Provider | active | <button>Edit</button> <button>Delete</button> |
| Provider Name | provider@example.com | Provider | active | <button>Edit</button> <button>Delete</button> |
| providernext | providernext@example.com | Provider | active | <button>Edit</button> <button>Delete</button> |
| customernew | customernew@example.com | Customer | active | <button>Edit</button> <button>Delete</button> |
| customer3 | customer123@example.com | Customer | banned | <button>Edit</button> <button>Delete</button> |
| provider123 | provider123@example.com | Provider | active | <button>Edit</button> <button>Delete</button> |
| customer 9 | customer123@example.com | Customer | banned | <button>Edit</button> <button>Delete</button> |

This is where I can manage orders

Orders

Create New Order

| Customer | Book Title | Order Date | Actions |
|---------------|------------|------------|---------|
| <div>12</div> | | | |