**Online Bus Booking System - Complete Documentation**

**1. Project Overview**

The **Online Bus Booking System** is a Java Full Stack web application developed using:

* **Backend:** Spring Boot 3+, Spring Data JPA, Spring Security, JWT
* **Frontend:** HTML, CSS, Thymeleaf, Bootstrap, JavaScript
* **Database:** MySQL (based on config)
* **Testing:** JUnit, Mockito

It allows users to register, log in, search buses, book seats, make payments, and give feedback. Admins can manage buses and view feedback.

**2. Features**

**User Features**

* Registration & Login (JWT-based authentication)
* Role-based redirection (User: search page, Admin: admin dashboard)
* Search buses by source and destination
* View booking history
* Profile management (update & change password)
* Submit feedback

**Admin Features**

* Admin Dashboard
* Add/Delete buses
* View all buses
* View user feedback

**3. Project Structure**

src/main/java/com/example/onlineBusBookingdemo/

├── controller/

├── entity/

├── repository/

├── service/

├── config/ <- Spring Security + JWT config

**Static Resources:**

src/main/resources/

├── static/

│ └── css/style.css

├── templates/

│ ├── register.html

│ ├── login.html

│ ├── search-buses.html

│ ├── search-result.html

│ ├── booking.html

│ ├── booking-success.html

│ ├── booking/history.html

│ ├── admin/add-bus.html

│ ├── admin/all-buses.html

│ ├── admin/admin-dashboard.html

│ ├── profile.html

│ └── feedback.html

**4. Entity Classes**

* Users: id, name, email, password, Set
* Role: id, name
* Bus: id, name, source, destination, date, time, pricePerSeat
* Booking: id, user, bus, date, travelDate, seatCount, totalAmount
* Feedback: id, user, bus, content, date

**5. Security & JWT**

* Configured Spring Security with JWT filter
* Users login with email/password
* On successful login, token is generated and stored client-side
* JwtAuthenticationFilter, JwtUtil, CustomUserDetailsService used
* Role-based redirection: Admin -> /admin-dashboard, User -> /buses/search

**6. Controllers**

* UserController: register, login, profile, password change
* BusController: search, add, delete, list buses
* BookingController: book seats, view booking history
* FeedbackController: submit and list feedbacks

**7. Services**

* UserService: register, authenticate, update profile, change password
* BusService: CRUD operations for Bus
* BookingService: booking and retrieval by user
* FeedbackService: save and list feedback

**8. Repository Interfaces**

* UserRepository
* RoleRepository
* BusRepository
* BookingRepository
* FeedbackRepository

All extend JpaRepository.

**9. Testing**

* Unit tests using JUnit & Mockito
* Tested Controllers, Services, and Repositories
* Mocked dependencies where needed (e.g., repositories)

**10. How to Run**

1. Clone the repository
2. Import into IntelliJ or Spring Tool Suite
3. Update application.properties or application.yml with DB config
4. Run OnlineBusBookingdemoApplication.java
5. Access via http://localhost:8080

**11. Future Enhancements**

* Email notification for bookings
* Online payment integration
* React frontend
* Admin analytics dashboard

**12. Technologies Used**

* Spring Boot
* Thymeleaf
* Bootstrap
* JPA/Hibernate
* JWT, Spring Security
* JUnit & Mockito
* Maven