

Module Objective

Students will learn:

- ✓ Authentication
 - ✓ Authorization
 - ✓ Role-Based Access Control (RBAC)
 - ✓ Spring Security configuration
 - ✓ JWT authentication
 - ✓ Role-based dashboard design
-

STEP 1: Create Spring Boot Project

✓ 1. Go to Spring Initializr

<https://start.spring.io>

Select:

- Project: **Maven**
 - Language: **Java**
 - Spring Boot: Latest stable
 - Group: `com.example`
 - Artifact: `fleetmanagement`
 - Java: 17
-

✓ 2. Add Dependencies

Add:

- Spring Web
- Spring Security
- Spring Data JPA
- MySQL Driver

Download → Extract → Open in IDE

✓ 3. Add JWT Dependencies Manually

Open pom.xml

Inside <dependencies> add:

```
<dependency>
    <groupId>io.jsonwebtoken</groupId>
    <artifactId>jjwt-api</artifactId>
    <version>0.11.5</version>
</dependency>

<dependency>
    <groupId>io.jsonwebtoken</groupId>
    <artifactId>jjwt-impl</artifactId>
    <version>0.11.5</version>
    <scope>runtime</scope>
</dependency>

<dependency>
    <groupId>io.jsonwebtoken</groupId>
    <artifactId>jjwt-jackson</artifactId>
    <version>0.11.5</version>
    <scope>runtime</scope>
</dependency>
```

Reload Maven.

STEP 2: Configure Database

Create database in MySQL:

```
CREATE DATABASE fleetdb;
```

Open application.properties

```
spring.datasource.url=jdbc:mysql://localhost:3306/fleetdb
spring.datasource.username=root
spring.datasource.password=yourpassword

spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
```

STEP 3: Create Role Enum

```
public enum Role {  
    ADMIN,  
    FLEET_MANAGER,  
    DRIVER,  
    CUSTOMER  
}
```

Explain:

- ENUM prevents invalid roles
 - Stores fixed role types
-

STEP 4: Create User Entity

```
@Entity  
public class User {  
  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    private Long id;  
  
    private String name;  
    private String email;  
    private String password;  
  
    @Enumerated(EnumType.STRING)  
    private Role role;  
}
```

Explain:

- `@Entity` → creates table
 - `@Enumerated` → stores role as text
-

STEP 5: Create Repository

```
public interface UserRepository extends JpaRepository<User, Long> {  
    Optional<User> findByEmail(String email);  
}
```

Explain:

- Used to fetch user during login
-

STEP 6: Implement Spring Security

Create SecurityConfig.java

```
@Configuration
@EnableWebSecurity
public class SecurityConfig {

    @Bean
    public SecurityFilterChain filterChain(HttpSecurity http) throws
Exception {
        http.csrf().disable()
            .authorizeHttpRequests(auth -> auth
                .requestMatchers("/admin/**").hasRole("ADMIN")
                .requestMatchers("/manager/**").hasRole("FLEET_MANAGER")
                .requestMatchers("/driver/**").hasRole("DRIVER")
                .requestMatchers("/customer/**").hasRole("CUSTOMER")
                .anyRequest().authenticated()
            )
            .formLogin();

        return http.build();
    }
}
```

Explain line-by-line:

- `.hasRole()` → restrict access
 - `.authenticated()` → login required
-

STEP 7: Design Login & Registration Screens

Login Screen UI



LOGIN

REGISTER

Sign in with:



or:

Email or username

Password

Remember me

[Forgot password?](#)

SIGN IN

[Not a member? Register](#)

Fields Required:

- Email
 - Password
 - Role dropdown
 - Login button
-

Registration Screen UI

The registration screen UI shows a 'Sign Up' form with the following fields:

- Name:
- E-Mail Address:
- Role:
- Password:
- Confirm Password:

A red banner at the top states: "The name field is required."

Sign Up

Name:

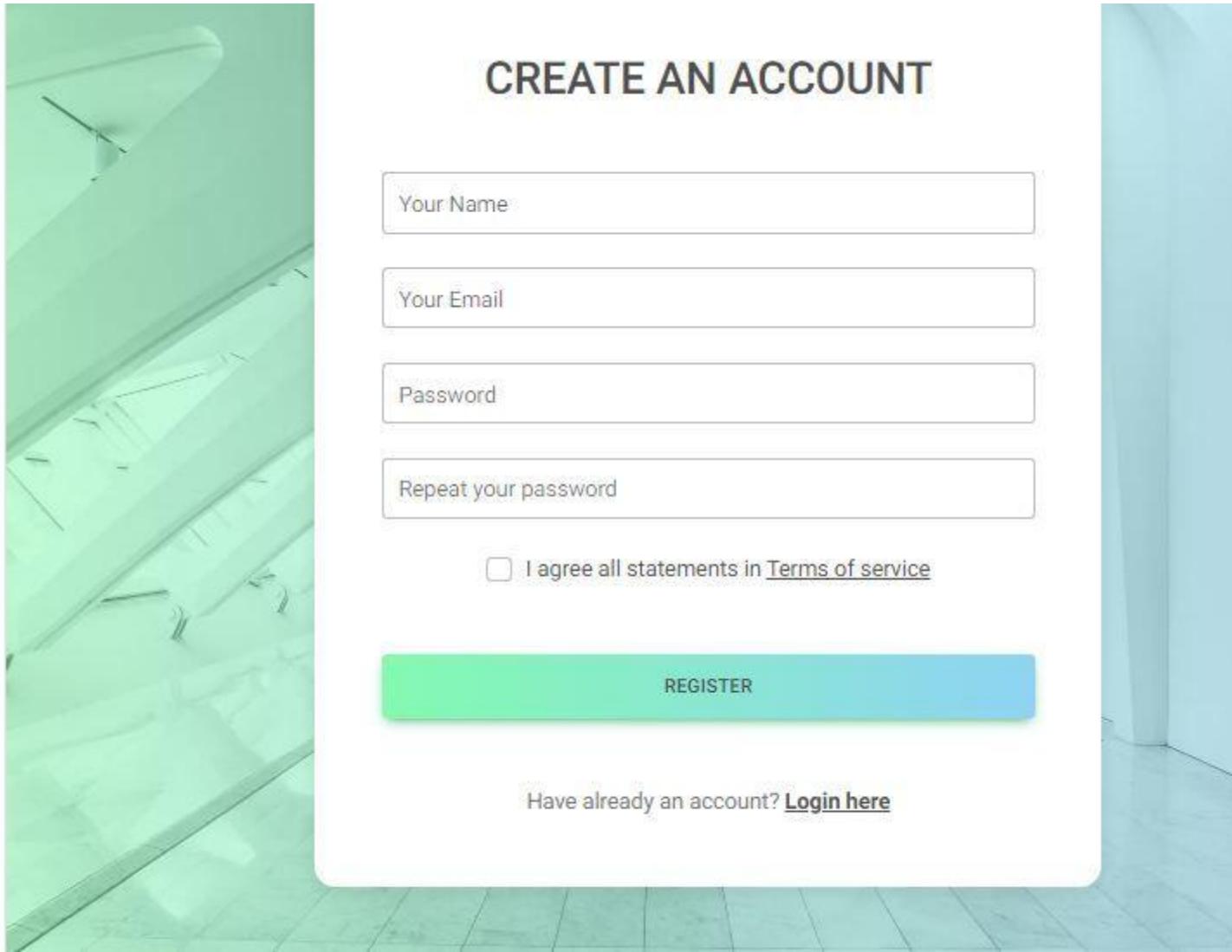
E-Mail Address:

Role:

Password:

Confirm Password:

Register Already have an account?



CREATE AN ACCOUNT

Your Name

Your Email

Password

Repeat your password

I agree all statements in [Terms of service](#)

REGISTER

Have already an account? [Login here](#)

4

Fields:

- Name
- Email
- Password
- Phone
- Role dropdown

Explain:

Role selected → stored in database.

STEP 8: Implement JWT Authentication Flow (Advanced)

Explain Flow:

```
User → Login API  
↓  
Validate using JPA  
↓  
Generate JWT Token  
↓  
Return Token  
↓  
Client sends token in Header
```

Header format:

```
Authorization: Bearer eyJhbGciOiJIUzI1NiIs...
```

STEP 9: Role-Based Dashboard Redirect

After successful login:

Check role:

```
if(role.equals("ROLE_ADMIN")) {  
    return "redirect:/admin/dashboard";  
}
```

Mapping:

Role	Redirect
ADMIN	/admin/dashboard
FLEET_MANAGER	/manager/dashboard
DRIVER	/driver/dashboard
CUSTOMER	/customer/dashboard

STEP 10: Create Dashboards

Admin Dashboard

 Home Health Reports Administration Settings

Reports

Sales report



Sales

1,000,000

Since last month

Total sales

Sales during Black Friday





Fleet Management



Dashboard >

Fin



Analytics ▾

\$

Finance

To

Vehicles

R

Trips

Ave



Vehicles >

\$



Trips >

\$12



Wheelzie

V 1.0

Wheelzie

Dashboard

Bookings

Units

Calendar

Clients

Drivers

Financials

Tracking

Messages

Dashboard



Total Revenue

\$8,450

↑ +2.86%

from last week



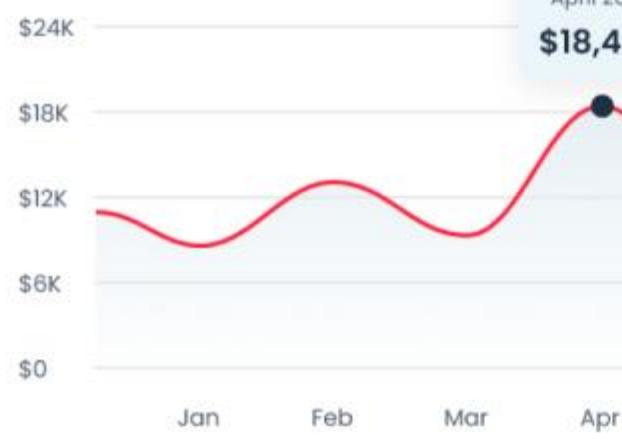
Rented Cars

214 Unit

↓ -2.86%

from last week

Earnings Summary



5

Metrics:

- Total Users
 - Total Bookings
 - Total Vehicles
 - Revenue
-

Fleet Manager Dashboard

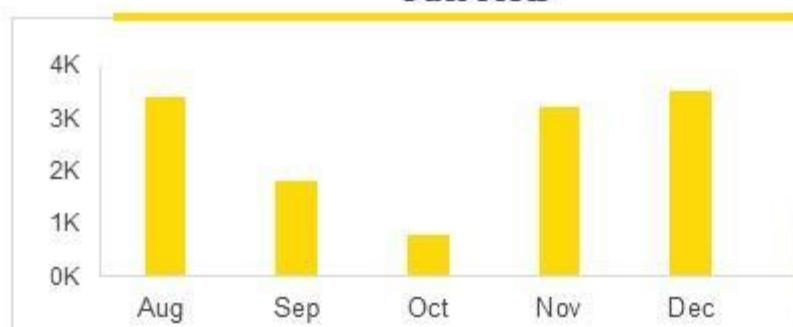
Fleet vehicle management system dashboard

This slide represents the dashboard representing the key metrics of the vehicle management system to monitor the vehicles and drivers in real-time.

Service Reminder



Fuel Costs



Vehicles



Open Issues



Louis mathews commented on service entry : #44
Repaired passenger size

Martha godwin commented on service entry : #36
Fixed cracked windshield

Robert watkins commented on issues: #160312
Add text here

Kellie linden commented on fuel entry : #229
Add text here

Juan miller commented on contact : #49
Add text here

This slide is 100% editable. Adapt it to your needs and capture the moment.



Golden

Delivering Received



The pair of sneacke

W94235675472RD

From:

San Diego

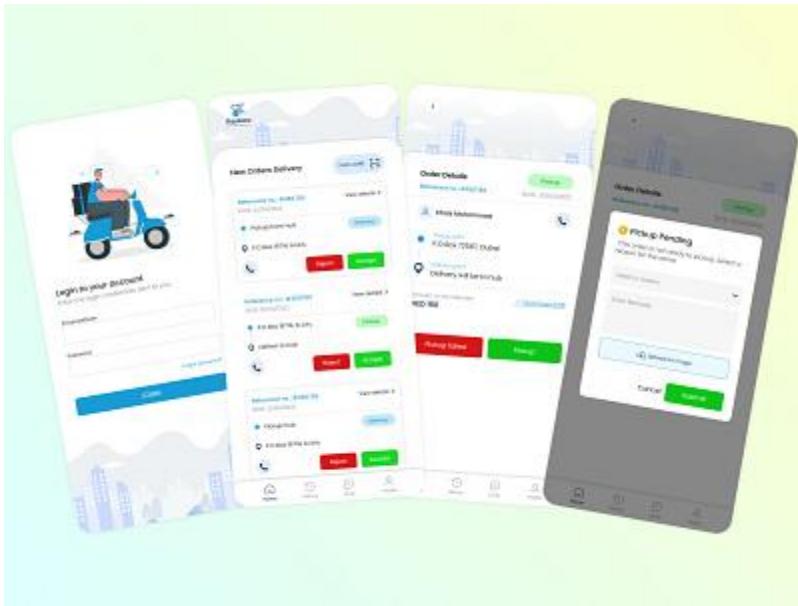
Status

On w

Metrics:

- Active Vehicles
- Drivers Available
- Trips Today

Driver Dashboard



Tracking

#127777489-DL-NY

Out for Delivery Parcel

- Package has left Courier Facility
SAN FRANCISCO, CALIFORNIA
 - Package arrived at Local Facility
NEW YORK CITY, NEW YORK
 - Out for Delivery
NEW YORK CITY, NEW YORK



Volkswagen Transporter

#127777489-DL-NY

In Transit Documents

-  **Package has left Courier Facility**
DETROIT, DENMARK

#127777489-DL-NY

In Customs Parcel

-  **Customs**
NEW YORK CITY, NEW YORK

101 Kg

Payload

51 km

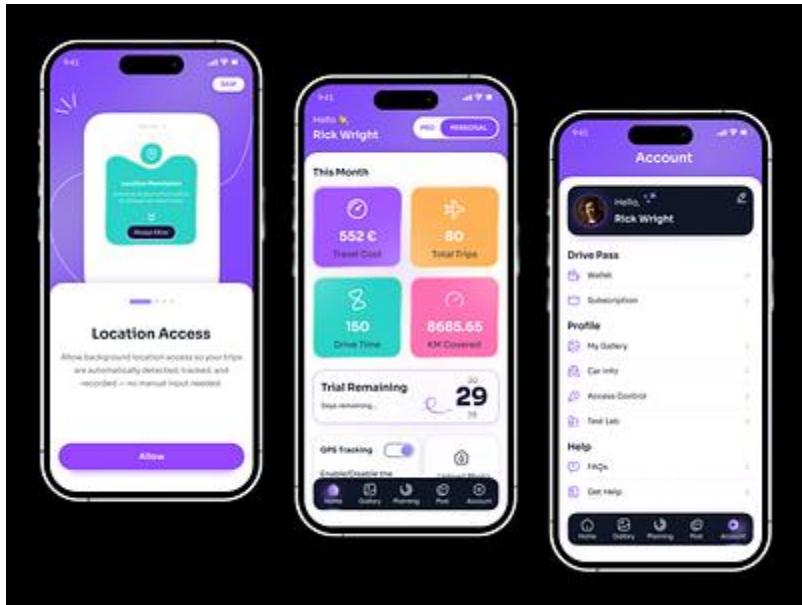
Distance

12

Volume

90

Estimated

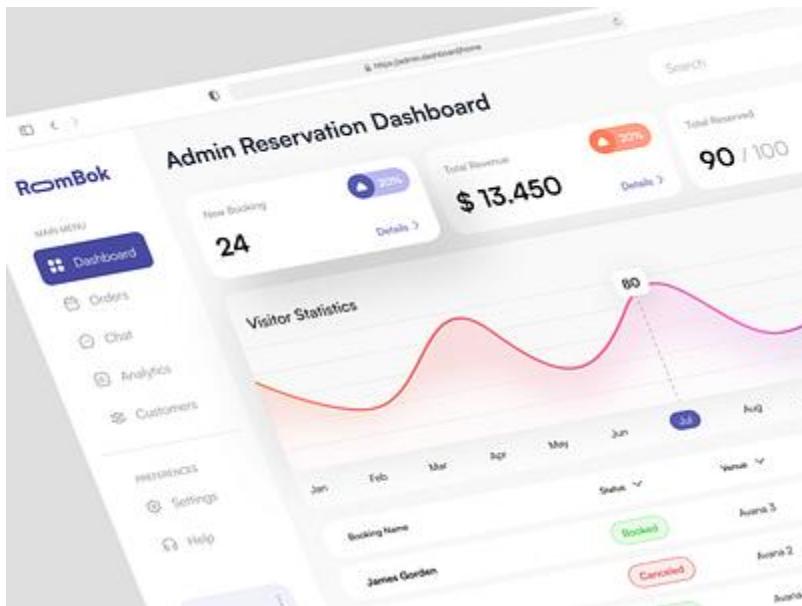


4

Metrics:

- Assigned Trips
- Trip Status
- Earnings

Customer Dashboard



The screenshot shows the CityCab application interface. At the top, there is a header with the logo 'CityCab' and the text 'Welcome Back, John!'. On the right side of the header is a user profile icon for 'John Migurdia' and a settings gear icon. Below the header is a navigation menu on the left with the following items: Dashboard, Bookings, Riders, Dispatchers, Drivers, Cars (which is the selected item, indicated by a green background), Reports, Tracking Bookings, Pricing Model, Feedback, and Payments. A large green button at the bottom of the menu says 'Create Booking' with a car icon. The main content area is titled 'Cars' and displays a grid of six car listings. Each listing includes a small image of the car, the model name, its category, and some basic statistics. There are also 'View More' links and 'On Trip' status indicators. The car models shown are Maruti Suzuki Swift (VXi) Mini, Toyota Innova (GX7) MPV, Hyundai Aura (SX-0) Sedan, Maruti Suzuki Ertiga (VXi) MPV, Maruti Suzuki WagonR (VXi) Mini, and Maruti Suzuki Celerio (VXi) Mini.

Car Model	Category	Statistics
Maruti Suzuki Swift (VXi)	Mini	4 passengers, Manual, Petrol
Toyota Innova (GX7)	MPV	6 passengers, Manual, Diesel
Hyundai Aura (SX-0)	Sedan	5 passengers, Manual, Petrol
Maruti Suzuki Ertiga (VXi)	MPV	6 passengers, Manual, CNG
Maruti Suzuki WagonR (VXi)	Mini	4 passengers, Manual, CNG
Maruti Suzuki Celerio (VXi)	Mini	4 passengers, Manual, CNG

4

Metrics:

- My Bookings
- Booking History
- Payment Status

STEP 11: Test Role Security

Try:

- Login as Driver
- Access /admin/dashboard

Expected:

✗ 403 Forbidden

Explain:

Authentication ≠ Authorization

STEP 12: Final Expected Output

Students must show:

- ✓ Working login page
- ✓ Role dropdown
- ✓ Role stored in DB
- ✓ Restricted endpoints
- ✓ Separate dashboards
- ✓ Dashboard metric tiles

Important Concepts to Emphasize

Ask students:

1. Difference between 401 & 403?
 - 401 → Not logged in
 - 403 → No permission
2. Why JWT is better than session?
 - Stateless
 - Scalable
 - Works with frontend frameworks
3. What is RBAC?
 - Access based on roles