

## **RideService.java**

```
public class RideService {  
    public String bookRide(String pickup,  
String dropoff, String time) {  
        if (pickup == null ||  
pickup.isEmpty() ||  
        dropoff == null ||  
dropoff.isEmpty() ||  
        time == null ||  
time.isEmpty()) {  
            return null; // invalid  
booking  
        }  
  
        return "BOOK" +  
System.currentTimeMillis();  
    }  
}
```

## **RideServiceTest.java**

```
import org.junit.jupiter.api.Test;  
import static  
org.junit.jupiter.api.Assertions.*;  
  
public class RideServiceTest {  
    @Test  
    void testValidBooking() {
```

```
        RideService service = new
RideService();
        String bookingId =
service.bookRide("UTD Campus", "DFW
Airport", "10:30 AM");
        assertNotNull(bookingId, "Booking
ID should not be null for valid input");
        assertTrue(bookingId.startsWith("BOOK"),
"Booking ID should start with
'BOOK'");
    }

    @Test
    void testInvalidBooking() {
        RideService service = new
RideService();
        String bookingId =
service.bookRide("", "DFW Airport",
"10:30 AM");
        assertNull(bookingId, "Booking ID
should be null for invalid input");
    }
}
```