**🧩 Prompt 1 — Phân tích hàm & Edge Case (bám đúng signature)**

**🎯 Vai trò**

Bạn là **Kỹ sư QA cao cấp** phân tích tính năng **Đặt xe (Booking)** của **RideNow** (Java Servlet/JSP, Maven, MVC).  
Mọi kiểm thử **không chạm DB** (mock toàn bộ DAO/Service phụ thuộc).

**📂 Yêu cầu thư mục**

Tạo file duy nhất:

cursor-output/booking-tests/docs/01\_analysis.md

**🎯 Hàm cần phân tích (KHÔNG đổi signature)**

**Service: IOrderService**

int bookOneBike(int customerId, int bikeId, Date start, Date end) throws Exception

boolean isBikeAvailable(int bikeId, Date start, Date end) throws SQLException

List<IOrderService.OverlappedRange> getOverlappingRanges(int bikeId, Date start, Date end) throws SQLException

**Service: IMotorbikeService**

List<Motorbike> findAll()

List<Motorbike> findByTypeId(int typeId)

List<MotorbikeListItem> search(Integer typeId, Date startDate, Date endDate, BigDecimal maxPrice, String keyword, String sort, int page, int size)

int count(Integer typeId, Date startDate, Date endDate, BigDecimal maxPrice, String keyword)

List<Motorbike> findAllByOwnerAccount(int accountId, String role)

MotorbikeListItem getDetail(int bikeId)

List<Motorbike> getByPartnerId(int partnerId)

**Servlet (mock Service + HttpServlet\*)**

BookingServlet.doPost(HttpServletRequest, HttpServletResponse)

MotorbikeDetailServlet.doGet(HttpServletRequest, HttpServletResponse)

MyOrdersServlet.doGet(HttpServletRequest, HttpServletResponse)

**🧱 Đầu ra**

Viết **2 bảng Markdown** (Service / Servlet) theo mẫu:

| **Function** | **Nhiệm vụ chính** | **Tham số (kiểu)** | **Giá trị trả** | **Quy tắc nghiệp vụ chính** | **Edge cases (≥3)** | **Cần mock** |
| --- | --- | --- | --- | --- | --- | --- |

**💡 Gợi ý Edge Case**

* Ngày: start ≥ end, null date, biên cùng ngày.
* Xe bận / overlap, ID không tồn tại, role không hợp lệ.
* Servlet: chưa đăng nhập, thiếu tham số, forward/redirect, 404 khi không thấy xe.

**🧩 Prompt 2 — Ma trận Test (Given – When – Then)**

**🎯 Vai trò**

Bạn là **Kỹ sư QA** thiết kế test case cho **Booking**.  
Không chạm DB, chỉ mock.

**📂 Yêu cầu thư mục**

Tạo file:

cursor-output/booking-tests/docs/02\_matrix.md

**🎯 Phạm vi**

* **Service:** toàn bộ hàm IOrderService & IMotorbikeService ở Prompt 1.
* **Servlet:** BookingServlet.doPost, MotorbikeDetailServlet.doGet, MyOrdersServlet.doGet.

**🎯 Coverage**

* **Service:** ≥3 case/hàm (Happy, Edge, Error)
* **Servlet:** ≥2 case/hàm (thành công + lỗi/thiếu dữ liệu/không đăng nhập)

**🧱 Bảng Markdown**

| **Test Case ID** | **Hàm** | **Loại (Happy/Edge/Error/State)** | **Given (Setup)** | **When (Hành động)** | **Then (Kết quả mong đợi)** | **Ưu tiên** |
| --- | --- | --- | --- | --- | --- | --- |

**⚠️ Lưu ý**

* **bookOneBike:** xe bận → Exception; ngày đảo; ID âm; return orderId > 0 khi thành công.
* **isBikeAvailable:** true/false; SQLException path.
* **getOverlappingRanges:** rỗng và ≥1 phần tử (kiểm tra orderId/start/end).
* **search/count:** lọc null, từ khóa rỗng, sort không hợp lệ, phân trang.
* **Servlet doPost:** param thiếu; chưa login; success redirect; fail forward + message.

**🧩 Prompt 3 — Sinh mã Test Class (JUnit 5 + Mockito)**

**🎯 Vai trò**

Bạn là **Lập trình viên Java cấp cao** tạo **test runnable** cho **RideNow Booking (Java Servlet/JSP, MVC)**.  
KHÔNG chạm DB – mock toàn bộ DAO/Service & Servlet API.

**📂 Thư mục xuất file**

Sinh các file Java vào:

AI\_Test\_Output/booking-tests/java-stubs/src/test/java/

Gợi ý package:  
booking.tests.\*, booking.utils.\*, booking.servlets.\*UnderTest

**🧩 Phụ thuộc (phải khớp POM)**

* JUnit 5 → **5.10.0**
* Mockito core → **5.5.0**
* Mockito JUnit Jupiter → **5.5.0**
* AssertJ → **3.24.2**
* Jakarta Servlet API → **6.0.0**

**🎯 Yêu cầu chung**

* @ExtendWith(MockitoExtension.class)
* Dùng @Mock, @InjectMocks
* Không import wildcard
* Code biên dịch được ngay
* Dùng AssertJ hoặc JUnit Assertions
* Mock: HttpServletRequest / Response / Session / RequestDispatcher
* Dữ liệu mẫu:
  + Date.valueOf("2025-10-25")
  + BigDecimal("150000.00")
* verify(times(1)) cho call quan trọng
* Mỗi @Test có @DisplayName ghi TC ID (map từ 02\_matrix.md)
* doGet / doPost là protected →
  + gọi bằng servlet.service(req, resp) với stub when(req.getMethod()).thenReturn("GET"/"POST"),
  + hoặc tạo bridge method doGetPublic / doPostPublic trong lớp \*UnderTest.

**🗂️ Tiện ích dùng lại – booking.utils.MockSetupFactory**

* stubSession(req, session)
* stubForward(req, path) → trả RequestDispatcher mock
* setPrivateField(target, "orderService"/"motorbikeService", mock)
* Factory mock domain: mockCustomer, mockBike, mockOrder
* Helper stub: isBikeAvailable, getDetail, search…

**📘 Các class test chính**

**1️⃣ OrderServiceTest**

* testBookOneBike\_ValidInput\_ShouldReturnOrderId()
* testBookOneBike\_BikeUnavailable\_ShouldThrow()
* testBookOneBike\_InvalidDates\_ShouldThrow()
* testIsBikeAvailable\_ShouldReturnTrueFalse\_AndHandleSQLException()
* testGetOverlappingRanges\_EmptyAndNonEmpty\_ShouldMapCorrectly()

**2️⃣ MotorbikeServiceTest**

* testFindAll\_ShouldReturnList()
* testFindByTypeId\_InvalidType\_ShouldReturnEmpty()
* testSearch\_WithFiltersAndPaging\_ShouldReturnExpectedSlice()
* testCount\_WithFilters\_ShouldReturnCorrectTotal()
* testGetDetail\_Found\_ShouldReturnItem()
* testGetDetail\_NotFound\_ShouldHandleGracefully()
* testFindAllByOwnerAccount\_RoleEdgeCases()
* testGetByPartnerId\_ShouldReturnPartnerBikes()

**3️⃣ Servlet Tests**

* **BookingServletTest**
  + testDoPost\_LoggedInAndValidParams\_ShouldRedirectSuccess()
  + testDoPost\_MissingDates\_ShouldForwardWithError()
  + testDoPost\_NotLoggedIn\_ShouldRedirectToLogin()
* **MotorbikeDetailServletTest**
  + testDoGet\_ExistingBike\_ShouldSetAttributesAndForward()
  + testDoGet\_InvalidBike\_ShouldSend404()
* **MyOrdersServletTest**
  + testDoGet\_LoggedInCustomer\_ShouldListOrders()
  + testDoGet\_NotLoggedIn\_ShouldRedirectToLogin()

**🎨 Gợi ý package (không trùng package main)**

package booking.tests.service;

package booking.tests.servlet;

package booking.utils;

package booking.servlets; // chứa \*UnderTest nếu cần bridge public

**✅ Ràng buộc thêm**

* KHÔNG ghi vào src/test/java của project chính.
* KHÔNG gọi DB/DAO thật; tất cả repository/service phụ thuộc đều phải mock.
* Test phải biên dịch và chạy được **độc lập trong runner POM**.

**🧩 Fix & Runner JaCoCo (JDK 23)**

**🎯 Vai trò**

Bạn là **Senior Java Dev** – cần sửa test & runner để bộ test Booking chạy pass và có **JaCoCo report**,  
**KHÔNG chạm vào source app chính**, chỉ sửa trong:

AI\_Test\_Output/booking-tests

**✅ Mục tiêu**

1. Fix JaCoCo bị lỗi với JDK 23 (Unsupported class file major version 67)
2. Sửa 3 test servlet fail do stub sai / verify lệch
3. Không đổi hành vi code under-test
4. Giữ nguyên cấu trúc thư mục hiện có

**⚙️ Việc cần làm theo file**

**1️⃣ runner/pom.xml**

Giữ nguyên build-helper-maven-plugin & maven-surefire-plugin.  
Thêm / điều chỉnh **maven-compiler-plugin** (release 17) và **JaCoCo 0.8.12**.

<build>

<plugins>

<!-- Giữ nguyên build-helper & surefire hiện có -->

<!-- Compiler: dùng --release 17 -->

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.11.0</version>

<configuration>

<release>17</release>

</configuration>

</plugin>

<!-- JaCoCo: 0.8.12 để hỗ trợ JDK 23 -->

<plugin>

<groupId>org.jacoco</groupId>

<artifactId>jacoco-maven-plugin</artifactId>

<version>0.8.12</version>

<executions>

<execution>

<goals>

<goal>prepare-agent</goal>

</goals>

</execution>

<execution>

<id>report</id>

<phase>test</phase>

<goals>

<goal>report</goal>

</goals>

</execution>

</executions>

</plugin>

</plugins>

</build>

**💼 Vai trò**

Bạn là **lập trình viên kiểm thử Java cấp cao** – sửa các test bị lỗi trong:

AI\_Test\_Output/booking-tests/java-stubs/src/test/java/booking/tests/servlet/

**📄 Cụ thể:**

* BookingServletTest.java
* MotorbikeDetailServletTest.java

**⚙️ Bối cảnh:**

Maven runner dùng **JUnit 5 + Mockito + AssertJ + JaCoCo**  
→ Test fail do verify sai hoặc stub sai tên param (không do code chính).  
**Không được chỉnh logic code under-test.**

**🔧 Yêu cầu sửa chi tiết**

**1️⃣ BookingServletTest.java**

Ở test **testDoPost\_MissingDates\_ShouldForwardWithError**:

* Dòng verify session.setAttribute(...) đang kỳ vọng anyString()  
  → Sửa thành nullable(String.class) để chấp nhận null.
* Giữ nguyên verify forward (RequestDispatcher.forward()).
* Không thay đổi logic mock khác.

📌 **Mẫu fix:**

verify(httpSession).setAttribute(eq("book\_error"), nullable(String.class));

verify(requestDispatcher, times(1)).forward(eq(request), eq(response));