**Test Plan for Rental Car booking Project**

**1) Test Plan Identifier**

* Unique ID: "RBCS\_TP\_001".
* Version control and tracking.
* Example: Rental Car Booking System Test Plan.
* Prepared by- tejas khairnar

**2) References**

* Requirements specification document.
* Design documents.
* User manuals or guides.
* Any third-party APIs used (e.g., payment gateway).

**3) Introduction**

Provide a brief overview of the rental car booking system, including the purpose of the system and why the test plan is necessary. Highlight key goals such as ensuring system functionality, usability, and performance.

**4) Test Items**

* Modules to test:
  1. User registration.
  2. Vehicle search.
  3. Booking functionality.
  4. Payment processing.
  5. Booking management (modifications/cancellations).

**5) Software Risk Issues**

* Performance issues during peak traffic.
* Payment system failures or security breaches.
* Data inconsistency in bookings.

**6) Features to be Tested**

* Login and user authentication.
* Vehicle availability and search functionality.
* Booking and confirmation flow.
* Payment gateway integration.
* Cancellations and refunds.

**7) Features not to be Tested**

* External service integration (e.g., Google Maps API).
* Compatibility with third-party booking sites.
* Minor UI elements not affecting functionality.

**8) Approach**

* **Unit testing** for individual modules.
* **Integration testing** for combined functionalities.
* **Manual testing** for usability and user interface checks.
* **Automated testing** for repetitive or regression testing.

**9) Item Pass/Fail Criteria**

* Pass: Expected result matches actual result.
* Fail: Any mismatch between expected and actual outcomes.
* Fail: Critical defects preventing further execution.

**10) Suspension Criteria and Resumption Requirements**

* Suspend: Critical bugs found that block testing.
* Resume: Bugs resolved and verified by the development team.
* Notify: Team for immediate attention on suspensions.

**11) Test Deliverables**

* **Test cases**: Detailed scenarios for all features.
* **Test logs**: Execution records.
* **Defect reports**: Documenting issues found.
* **Test summary**: Final report on testing results.

**12) Remaining Test Tasks**

* Outline any remaining tasks that need to be completed. For example, regression testing after bug fixes or final validation after integration of all system components.

**13) Environmental Needs**

* List the hardware and software environments required for testing. This includes specific operating systems, browsers (e.g., Chrome, Firefox), database configurations, network settings, and any necessary test tools.

**14) Staffing and Training Needs**

* **Testers**: 3-4 testers with manual and automation skills.
* **Training**: On test tools (e.g., Selenium).
* **Specialists**: Security testing or performance testing expertise.

**15) Responsibilities**

* **Test Manager**: Oversees test planning and execution.
* **Test Engineers**: Write and execute test cases.
* **Developers**: Address defects found during testing.
* **Project Manager**: Reviews and signs off on the test plan.

**16) Schedule**

* **Test Design**: From October 1st to October 5th.
* **Test Execution**: October 6th to October 15th.
* **Defect Fixing & Retesting**: October 16th to October 20th.
* **Final Testing & Sign-off**: October 21st.

**17) Planning Risks and Contingencies**

* **Risk**: Delays in module completion.
* **Risk**: Resource shortages (e.g., testers unavailable).
* **Contingency**: Adjust testing timelines, reallocate resources.

**18) Approvals**

* **Project Manager**: Final approval.
* **Test Lead**: Test strategy validation.
* **Developers**: Confirmation on test scope and fixes.

**19) Glossary**

* **Test Case**: A specific scenario to be tested.
* **Defect**: An issue or bug found during testing.
* **Regression Testing**: Testing to ensure new code changes don’t break existing functionality.
* **UAT**: User Acceptance Testing.