Session9 Assignment(penetration testing methodologies)

1.Table about characteristics, advantages, and disadvantages of each methodology:

	White Box	**Black Box**	**Grey Box**
characteristics	- Complete knowledge of the internal structure, design, and code. - Analyzes the internal logic, data flows, and decision points. - every detail is accessible.	- No knowledge of the internal workings. - Tests or analyses are based on inputs and expected outputs. - Internal mechanisms are hidden or treated as unknown.	- Partial knowledge of the internal workings. - Combines elements of black and white box testing, using partial insights to guide testing. - Some aspects are known while others are treated as opaque.
advantages	- testing is thorough as the entire code and structures are tested. - It results in the optimization of code removing errors and helps in removing extra lines of code. - Early Detection of Defects - Integration with SDLC - Detection of Complex Defects - Comprehensive Test Cases	- The tester does not need to have more functional knowledge or programming skills - It is efficient for implementing the tests in the larger system. - Tests are executed from the user's or client's point of view. - Test cases are easily reproducible. - It is used to find the ambiguity and contradictions in the functional specifications.	- Users and developers have clear goals while doing testing. - testing is mostly done by the user perspective. - Testers are not required to have high programming skills for this testing. - Overall quality of the product is improved. - developers have more time for defect fixing. - Benefits of black box and white box testing - avoids conflicts between a tester and a developer. - much more effective in integration testing.

Session9 Assignment(penetration testing methodologies)

disadvantages	- need to have programming knowledge and access to the source code to perform tests. - Testers may focus too much on the internal workings of the software and may miss external issues. - Testers are required to have in-depth knowledge of the code and programming language. - Missing functionalities cannot be detected as the code that exists is tested.	- There is a possibility of repeating the same tests while implementing the testing process. - It is difficult to execute the test cases because of complex inputs at different stages of testing. - Sometimes, the reason for the test failure cannot be detected. - Some programs in the application are not tested. - It does not reveal the errors in the control structure.	- Difficulty in defect association - Limited access to internal structure leads to limited access for code path traversal. - Source code not accessible - Not suitable for algorithm testing - Most of the test cases are difficult to design.

2.Real-World Scenarios for each methodology:

1. Black Box

Scenario: Testing a Mobile Application's User Interface (UI)

WHY! The tester doesn’t need to know how the app’s code or algorithms are implemented.

Example: Testing whether a shopping app correctly processes credit card payments and generates confirmation messages.

2. White Box

Scenario: Debugging Software for Security Vulnerabilities

WHY! Developers or testers analyze the code to identify logical errors, security loopholes (e.g., SQL injection points), or inefficient algorithms.

Example: Examining the backend code of an online banking system to ensure encryption methods are implemented correctly.

3. Gray Box

Scenario: Testing a Web Application with Limited API Documentation

WHY! Testers have partial knowledge, such as API endpoints and their expected behaviors, but lack access to the full source code.

Combines external functional testing (black box) with some insights into the system's workings to test integration and identify edge cases.

Example: Verifying how a payment gateway handles invalid inputs like expired cards or incorrect CVVs by combining functional testing with known API details.