

1. Налаштування Jenkins

- Завантажено та встановлено **Jenkins** на локальний сервер.
- Виконано **базове налаштування**, включаючи встановлення плагінів:
 - **Pipeline**
 - **Allure Reports**
 - **Parameterized Trigger**
 - **HTML Publisher**
- Створено користувачів та призначено ролі для налаштування доступу.

2. Створення завдань

а. Завдання для простого методу main

- **Мета:** Перевірка виконання простого Java-коду з методом main.
- **Кроки:**
 - Створено **Freestyle Project**.
 - Додано параметр **String Parameter** для введення тестових даних.

У розділі **Build** налаштовано виконання Java-коду:

```
javac Main.java
```

```
java Main ${param1}
```

Логуювання та виведення результатів налаштовані.

б. Завдання для API

- **Мета:** Виконання API-тестів з параметризованими даними.
- **Кроки:**
 - Створено **Freestyle Project**.
 - Параметризовано вхідні дані для тестового сценарію.
 - Вказано шлях до тестового коду API (Postman або REST Assured).

Налаштовано запуск через **Maven**:

```
mvn test -Dtest="ApiTestSuite" -Dparam1=${param1}
```

-
- Інтегровано **Allure Reports** для генерації звітів.

с. Завдання для UI (+Allure)

- **Мета:** Виконання автоматизованих UI-тестів з генерацією Allure-звіту.
- **Кроки:**
 - Створено **Pipeline Job**.

У **Jenkinsfile** описано скрипт:

```
pipeline {  
    agent any  
  
    stages {  
        stage('Build') {  
            steps {  
                sh 'mvn clean compile'  
            }  
        }  
  
        stage('Test') {  
            steps {  
                sh 'mvn test -Dtest="UITestSuite"'  
            }  
        }  
  
        stage('Allure Report') {  
            steps {  
                allure([results: [[path: '**/allure-results']],  
reportBuildPolicy: 'ALWAYS'])  
            }  
        }  
    }  
}
```

-
- Додано звіт **Allure**.

d. Завдання для тестування продуктивності

- **Мета:** Виконання JMeter тестів.
- **Кроки:**
 - Створено **Freestyle Project**.

Додано виконання **JMeter**-скрипта:

```
jmeter -n -t Test_Scenario.jmx -l results.jtl
```

-
- Налаштовано збереження результатів у **HTML-звіт**.

e. Завдання для мобільних додатків

- **Мета:** Виконання тестів мобільних додатків.
- **Кроки:**
 - Створено **Pipeline Job**.

У **Jenkinsfile** описано виконання мобільних тестів:

```
pipeline {  
  
    agent any  
  
    stages {  
  
        stage('Build APK') {  
  
            steps {  
  
                sh './gradlew assembleDebug'  
  
            }  
  
        }  
  
        stage('Test') {  
  
            steps {  
  
                sh 'mvn test -Dtest="MobileTestSuite"'  
  
            }  
  
        }  
  
        stage('Allure Report') {
```

```

        steps {
            allure([results: [[path: '**/allure-results']],
reportBuildPolicy: 'ALWAYS'])
        }
    }
}
}
}

```

○

f. Завдання для настільних додатків

- **Мета:** Автоматизація тестів для настільних додатків.
- **Кроки:**
 - Створено **Pipeline Job**.

У **Jenkinsfile** описано виконання **Winium-тестів**:

```

pipeline {
    agent any

    stages {
        stage('Test') {
            steps {
                sh 'java -jar DesktopTests.jar'
            }
        }

        stage('Allure Report') {
            steps {
                allure([results: [[path: '**/allure-results']],
reportBuildPolicy: 'ALWAYS'])
            }
        }
    }
}

```

```
}  
  
}  
  
}  
  
}
```

Висновки:

- Усі завдання налаштовані для виконання різних типів тестів (Java, API, UI, мобільні додатки, продуктивність, настільні додатки) з підтримкою параметризації та інтеграцією **Allure** для звітності.