BASETEST:

```
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.AfterTest;
import org.testng.annotations.BeforeTest;

public class BaseTest {
	ChromeDriver driver;

		@BeforeTest
		public void openApp() {
			driver = new ChromeDriver();
			driver.manage().window().maximize();
			driver.get("http://localhost:9010/");
		}

		@AfterTest
		public void closeApp() {
		driver.quit();
		}

}
```

REGISTRATION PAGE:

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;

public class RegistrationPage {

	@FindBy(linkText = "New User? Register Here")
	private WebElement regBTN;

	@FindBy(id = "name")
	private WebElement inputNameBox;

	@FindBy(id = "email")
	private WebElement inputEmailBox;

	@FindBy(id = "password")
	private WebElement inputPasswordBox;

	@FindBy(xpath = "//button[contains(@type,'submit')]")
	private WebElement regiBTN;

	public RegistrationPage(WebDriver driver) {
		PageFactory.initElements(driver, this);
	}

	public void regPage() {
		regBTN.click();
	}

	public void inputName() {
        inputNameBox.sendKeys("sfEFWasfzfarfa");
    }

    public void inputEmail() {
        inputEmailBox.sendKeys("qwer@gmail.com");
    }

    public void inputPassword() {
        inputPasswordBox.sendKeys("qwer");
    }

    public void clickOnRegBTN() {
        regiBTN.click();
    }


}
```

LOGIN PAGE:

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;

public class LogInPage {

    @FindBy(id = "email")
    private WebElement inputEmailBox;

    @FindBy(id = "password")
    private WebElement inputPasswordBox;

    @FindBy(xpath = "//button[contains(@type,'submit')]")
    private WebElement LogInBTN;

    public LogInPage(WebDriver driver) {
        PageFactory.initElements(driver, this);
    }

    public void inputEmail() {
        inputEmailBox.sendKeys("qwer@gmail.com");
    }

    public void inputPassword() {
        inputPasswordBox.sendKeys("qwer");
    }

    public void clickOnLogInBTN() {
        LogInBTN.click();
    }

}
```

HOMEPAGE:

```
import java.time.Duration;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.interactions.Actions;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.openqa.selenium.support.ui.WebDriverWait;

public class HomePage {

	@FindBy(xpath = "(//a[contains(@class,'btn btn-primary')])[1]")
	private WebElement firstItem;

	@FindBy(xpath = "(//a[contains(@class,'btn btn-primary')])[2]")
	private WebElement secItem;

	@FindBy(linkText = "Logout")
	private WebElement logoutBTN;

	JavascriptExecutor js;
	WebDriverWait wait;
	Actions action;


	public HomePage(WebDriver driver) {
		js = (JavascriptExecutor) driver;
		PageFactory.initElements(driver, this);
		action = new Actions(driver);
		wait = new WebDriverWait(driver, Duration.ofSeconds(60));
	}

	public void addFirstItem() throws InterruptedException {
		Thread.sleep(2000);
		js.executeScript("arguments[0].scrollIntoView();", firstItem);
		js.executeScript("arguments[0].click()",firstItem);
	}

	public void addSecondItem() {
		action.moveToElement(secItem).click().perform();
	}

	public void clickOnLogoutBTN() {
		logoutBTN.click();
	}

}
```

CARTPAGE:

```
import java.time.Duration;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.interactions.Actions;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.openqa.selenium.support.ui.WebDriverWait;

public class CartPage {

    @FindBy(xpath = "(//a[contains(@class,'nav-link')])[1]")
    private WebElement cart;

    @FindBy(xpath = "(//a[contains(@class,'btn btn-primary')])[1]")
    private WebElement placeOrdBTN;

    @FindBy(xpath = "(//a[contains(@class,'nav-link')])[1]")
    private WebElement homeBTN;

    JavascriptExecutor js;
    WebDriverWait wait;
    Actions action;

    public CartPage (WebDriver driver) {
        js = (JavascriptExecutor) driver;
        PageFactory.initElements(driver, this);
        action = new Actions(driver);
        wait = new WebDriverWait(driver, Duration.ofSeconds(60));
    }

    public void goToCart() {
        cart.click();
    }

    public void clickOnPlaceOrdBTN() throws InterruptedException {
        Thread.sleep(2000);
        js.executeScript("arguments[0].scrollIntoView();", placeOrdBTN);
        js.executeScript("arguments[0].click()",placeOrdBTN);
    }

    public void clickOnHomeBTN() {
        homeBTN.click();
    }

}
```

ORDERCONFIRMATION PAGE:

```
package com.sportyshoes.pages;


import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.support.FindBy;

import org.openqa.selenium.support.PageFactory;
```

```
public class OrderConfirmPage {


	@FindBy(xpath = "/html/body/div[3]/div/p")
	private WebElement orderPlacedMsg;


	@FindBy(xpath = "/html/body/div[3]/div/p")
	private WebElement itemAddedMsg;


	public OrderConfirmPage(WebDriver driver) {
		PageFactory.initElements(driver, this);
	}


	public String getMsg() {
		return orderPlacedMsg.getText();
	}


	public String getMsg1() {
		return itemAddedMsg.getText();
	}
}
```

API TESTING WITH REST-ASSURED:

```
import org.testng.annotations.Test;


import io.restassured.RestAssured;


public class Get_Post_On_SportyShoes extends TestBase{


	@Test(priority=0)
	public void getAllProd() {
		RestAssured
```

```
                    .when()
                              .get("http://localhost:9010/get-shoes")
                    .then()
                              .assertThat()
                              .statusCode(200)
                              .log().all();

          logger.info("All Product Details fetched Successfully");
     }

     @Test(priority=1)
     public void getAllUsers() {
          RestAssured
                    .when()
                              .get("http://localhost:9010/get-users")
                    .then()
                              .assertThat()
                              .statusCode(200)
                              .log().all();

          logger.info("All Users Details fetched Successfully");
     }

     @Test(priority=2)
     public void addProd() {
          RestAssured
                    .when()
                              .post("http://localhost:9010/add-
shoe?id=101&image=image_url&name=SampleShoe&category=Running&sizes=9&price=1000")
                    .then()
                              .assertThat()
```

```
                                        .statusCode(200)
                                        .log().all();

                    logger.info("Add one Product Successfully");
        }

}
```

LOAD TESTING WITH JMETER:

(WITH 50 USERS)

```
<?xml version="1.0" encoding="UTF-8"?>
<jmeterTestPlan version="1.2" properties="5.0" jmeter="5.6.2">
  <hashTree>
    <TestPlan guiclass="TestPlanGui" testclass="TestPlan" testname="SportyShoes" enabled="true">
      <stringProp name="TestPlan.comments">This test plan was created by the BlazeMeter converter v.3.1.23. Please contact support@blazemeter.com for further support.</stringProp>
      <boolProp name="TestPlan.functional_mode">false</boolProp>
      <boolProp name="TestPlan.serialize_threadgroups">false</boolProp>
      <elementProp name="TestPlan.user_defined_variables" elementType="Arguments">
        <collectionProp name="Arguments.arguments"/>
      </elementProp>
      <stringProp name="TestPlan.user_define_classpath"></stringProp>
    </TestPlan>
    <hashTree>
      <HeaderManager guiclass="HeaderPanel" testclass="HeaderManager" testname="HTTP Header manager" enabled="true">
        <collectionProp name="HeaderManager.headers">
          <elementProp name="sec-ch-ua" elementType="Header">
            <stringProp name="Header.name">sec-ch-ua</stringProp>
            <stringProp name="Header.value">&quot;Chromium&quot;;v=&quot;116&quot;, &quot;Not)A;Brand&quot;;v=&quot;24&quot;, &quot;Google Chrome&quot;;v=&quot;116&quot;</stringProp>
          </elementProp>
          <elementProp name="sec-ch-ua-mobile" elementType="Header">
            <stringProp name="Header.name">sec-ch-ua-mobile</stringProp>
            <stringProp name="Header.value">?0</stringProp>
          </elementProp>
          <elementProp name="Accept" elementType="Header">
            <stringProp name="Header.name">Accept</stringProp>
            <stringProp name="Header.value">text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7</stringProp>
          </elementProp>
          <elementProp name="Upgrade-Insecure-Requests" elementType="Header">
            <stringProp name="Header.name">Upgrade-Insecure-Requests</stringProp>
            <stringProp name="Header.value">1</stringProp>
          </elementProp>
          <elementProp name="sec-ch-ua-platform" elementType="Header">
            <stringProp name="Header.name">sec-ch-ua-platform</stringProp>
            <stringProp name="Header.value">&quot;Windows&quot;</stringProp>
          </elementProp>
          <elementProp name="User-Agent" elementType="Header">
            <stringProp name="Header.name">User-Agent</stringProp>
            <stringProp name="Header.value">Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Safari/537.36</stringProp>
          </elementProp>
          <elementProp name="Sec-Fetch-Dest" elementType="Header">
            <stringProp name="Header.name">Sec-Fetch-Dest</stringProp>
            <stringProp name="Header.value">document</stringProp>
          </elementProp>
          <elementProp name="Sec-Fetch-User" elementType="Header">
            <stringProp name="Header.name">Sec-Fetch-User</stringProp>
```

```
            <stringProp name="Header.value">?1</stringProp>
          </elementProp>
          <elementProp name="Sec-Fetch-Mode" elementType="Header">
            <stringProp name="Header.name">Sec-Fetch-Mode</stringProp>
            <stringProp name="Header.value">navigate</stringProp>
          </elementProp>
        </collectionProp>
      </HeaderManager>
      <hashTree/>
      <Arguments guiclass="ArgumentsPanel" testclass="Arguments" testname="User Defined Variables" enabled="true">
        <collectionProp name="Arguments.arguments">
          <elementProp name="BASE_URL_1" elementType="Argument">
            <stringProp name="Argument.name">BASE_URL_1</stringProp>
            <stringProp name="Argument.value">localhost</stringProp>
            <stringProp name="Argument.metadata">=</stringProp>
          </elementProp>
        </collectionProp>
      </Arguments>
      <hashTree/>
      <ConfigTestElement guiclass="HttpDefaultsGui" testclass="ConfigTestElement" testname="HTTP Request Defaults" enabled="true">
        <elementProp name="HTTPsampler.Arguments" elementType="Arguments" guiclass="HTTPArgumentsPanel" testclass="Arguments" enabled="true">
          <collectionProp name="Arguments.arguments"/>
        </elementProp>
        <boolProp name="HTTPSampler.image_parser">true</boolProp>
        <boolProp name="HTTPSampler.concurrentDwn">true</boolProp>
        <stringProp name="HTTPSampler.concurrentPool">6</stringProp>
      </ConfigTestElement>
      <hashTree/>
      <DNSCacheManager guiclass="DNSCachePanel" testclass="DNSCacheManager" testname="DNS Cache Manager" enabled="true">
        <collectionProp name="DNSCacheManager.servers"/>
        <boolProp name="DNSCacheManager.clearEachIteration">true</boolProp>
        <boolProp name="DNSCacheManager.isCustomResolver">false</boolProp>
      </DNSCacheManager>
      <hashTree/>
      <AuthManager guiclass="AuthPanel" testclass="AuthManager" testname="HTTP Authorization Manager" enabled="true">
        <collectionProp name="AuthManager.auth_list"/>
        <boolProp name="AuthManager.controlledByThreadGroup">false</boolProp>
      </AuthManager>
      <hashTree/>
      <CookieManager guiclass="CookiePanel" testclass="CookieManager" testname="HTTP Cookie Manager" enabled="true">
        <collectionProp name="CookieManager.cookies"/>
        <boolProp name="CookieManager.clearEachIteration">true</boolProp>
        <boolProp name="CookieManager.controlledByThreadGroup">false</boolProp>
      </CookieManager>
      <hashTree/>
      <CacheManager guiclass="CacheManagerGui" testclass="CacheManager" testname="HTTP Cache Manager" enabled="true">
        <boolProp name="clearEachIteration">true</boolProp>
        <boolProp name="useExpires">false</boolProp>
        <boolProp name="CacheManager.controlledByThread">false</boolProp>
      </CacheManager>
      <hashTree/>
      <ThreadGroup guiclass="ThreadGroupGui" testclass="ThreadGroup" testname="Thread Group" enabled="true">
        <stringProp name="ThreadGroup.on_sample_error">continue</stringProp>
        <elementProp name="ThreadGroup.main_controller" elementType="LoopController" guiclass="LoopControlPanel" testclass="LoopController" enabled="true">
          <stringProp name="LoopController.loops">1</stringProp>
          <boolProp name="LoopController.continue_forever">false</boolProp>
        </elementProp>
        <stringProp name="ThreadGroup.num_threads">50</stringProp>
        <stringProp name="ThreadGroup.ramp_time">2</stringProp>
        <boolProp name="ThreadGroup.scheduler">false</boolProp>
        <stringProp name="ThreadGroup.duration">0</stringProp>
        <stringProp name="ThreadGroup.delay">0</stringProp>
        <boolProp name="ThreadGroup.delayedStart">false</boolProp>
        <boolProp name="ThreadGroup.same_user_on_next_iteration">true</boolProp>
      </ThreadGroup>
      <hashTree>
        <TransactionController guiclass="TransactionControllerGui" testclass="TransactionController"
testname="Test" enabled="true">
          <boolProp name="TransactionController.includeTimers">false</boolProp>
          <boolProp name="TransactionController.parent">false</boolProp>
        </TransactionController>
        <hashTree>
          <HTTPSamplerProxy guiclass="HttpTestSampleGui" testclass="HTTPSamplerProxy"
testname="http://localhost:9010/" enabled="true">
            <boolProp name="HTTPSampler.postBodyRaw">false</boolProp>
            <elementProp name="HTTPsampler.Arguments" elementType="Arguments"
guiclass="HTTPArgumentsPanel" testclass="Arguments" enabled="true">
              <collectionProp name="Arguments.arguments"/>
            </elementProp>
            <stringProp name="HTTPSampler.domain">${BASE_URL_1}</stringProp>
            <stringProp name="HTTPSampler.port">9010</stringProp>
            <stringProp name="HTTPSampler.protocol">http</stringProp>
            <stringProp name="HTTPSampler.method">GET</stringProp>
            <boolProp name="HTTPSampler.follow_redirects">true</boolProp>
            <boolProp name="HTTPSampler.auto_redirects">false</boolProp>
            <boolProp name="HTTPSampler.use_keepalive">true</boolProp>
            <boolProp name="HTTPSampler.DO_MULTIPART_POST">false</boolProp>
            <boolProp name="HTTPSampler.BROWSER_COMPATIBLE_MULTIPART">false</boolProp>
            <boolProp name="HTTPSampler.image_parser">false</boolProp>
            <boolProp name="HTTPSampler.concurrentDwn">false</boolProp>
            <stringProp name="HTTPSampler.concurrentPool">6</stringProp>
            <boolProp name="HTTPSampler.md5">false</boolProp>
            <intProp name="HTTPSampler.ipSourceType">0</intProp>
          </HTTPSamplerProxy>
          <hashTree>
            <HeaderManager guiclass="HeaderPanel" testclass="HeaderManager" testname="HTTP Header manager" enabled="true">
              <collectionProp name="HeaderManager.headers">
                <elementProp name="Sec-Fetch-Site" elementType="Header">
                  <stringProp name="Header.name">Sec-Fetch-Site</stringProp>
                  <stringProp name="Header.value">none</stringProp>
                </elementProp>
              </collectionProp>
            </HeaderManager>
            <hashTree/>
            <UniformRandomTimer guiclass="UniformRandomTimerGui" testclass="UniformRandomTimer" testname="Uniform Random Timer" enabled="true">
              <stringProp name="RandomTimer.range">0.0</stringProp>
              <stringProp name="ConstantTimer.delay">0</stringProp>
              <stringProp name="TestPlan.comments">Recorded time was 0 milliseconds</stringProp>
            </UniformRandomTimer>
            <hashTree/>
          </hashTree>
          <HTTPSamplerProxy guiclass="HttpTestSampleGui" testclass="HTTPSamplerProxy" testname="http://localhost:9010/register" enabled="true">
            <boolProp name="HTTPSampler.postBodyRaw">false</boolProp>
            <elementProp name="HTTPsampler.Arguments" elementType="Arguments" guiclass="HTTPArgumentsPanel" testclass="Arguments" enabled="true">
              <collectionProp name="Arguments.arguments"/>
            </elementProp>
            <stringProp name="HTTPSampler.domain">${BASE_URL_1}</stringProp>
            <stringProp name="HTTPSampler.port">9010</stringProp>
            <stringProp name="HTTPSampler.protocol">http</stringProp>
            <stringProp name="HTTPSampler.path">register</stringProp>
            <stringProp name="HTTPSampler.method">GET</stringProp>
            <boolProp name="HTTPSampler.follow_redirects">true</boolProp>
            <boolProp name="HTTPSampler.auto_redirects">false</boolProp>
            <boolProp name="HTTPSampler.use_keepalive">true</boolProp>
            <boolProp name="HTTPSampler.DO_MULTIPART_POST">false</boolProp>
            <boolProp name="HTTPSampler.BROWSER_COMPATIBLE_MULTIPART">false</boolProp>
            <boolProp name="HTTPSampler.image_parser">false</boolProp>
            <boolProp name="HTTPSampler.concurrentDwn">false</boolProp>
            <stringProp name="HTTPSampler.concurrentPool">6</stringProp>
            <boolProp name="HTTPSampler.md5">false</boolProp>
            <intProp name="HTTPSampler.ipSourceType">0</intProp>
          </HTTPSamplerProxy>
          <hashTree>
            <HeaderManager guiclass="HeaderPanel" testclass="HeaderManager" testname="HTTP
  Header manager" enabled="true">
              <collectionProp name="HeaderManager.headers">
                <elementProp name="Sec-Fetch-Site" elementType="Header">
                  <stringProp name="Header.name">Sec-Fetch-Site</stringProp>
                  <stringProp name="Header.value">same-origin</stringProp>
                </elementProp>
              </collectionProp>
            </HeaderManager>
            <hashTree/>
            <UniformRandomTimer guiclass="UniformRandomTimerGui"
  testclass="UniformRandomTimer" testname="Uniform Random Timer" enabled="true">
              <stringProp name="RandomTimer.range">1947.0</stringProp>
              <stringProp name="ConstantTimer.delay">973.5</stringProp>
              <stringProp name="TestPlan.comments">Recorded time was 1947 milliseconds</stringProp>
            </UniformRandomTimer>
            <hashTree/>
          </hashTree>
          <HTTPSamplerProxy guiclass="HttpTestSampleGui" testclass="HTTPSamplerProxy"
  testname="http://localhost:9010/register-user" enabled="true">
            <boolProp name="HTTPSampler.postBodyRaw">false</boolProp>
            <elementProp name="HTTPsampler.Arguments" elementType="Arguments"
  guiclass="HTTPArgumentsPanel" testclass="Arguments" enabled="true">
              <collectionProp name="Arguments.arguments">
                <elementProp name="password" elementType="HTTPArgument">
                  <boolProp name="HTTPArgument.always_encode">true</boolProp>
                  <stringProp name="Argument.name">password</stringProp>
                  <stringProp name="Argument.value">satya1</stringProp>
                  <stringProp name="Argument.metadata">=</stringProp>
                </elementProp>
                <elementProp name="name" elementType="HTTPArgument">
                  <boolProp name="HTTPArgument.always_encode">true</boolProp>
                  <stringProp name="Argument.name">name</stringProp>
                  <stringProp name="Argument.value">satya</stringProp>
                  <stringProp name="Argument.metadata">=</stringProp>
                </elementProp>
                <elementProp name="email" elementType="HTTPArgument">
                  <boolProp name="HTTPArgument.always_encode">true</boolProp>
                  <stringProp name="Argument.name">email</stringProp>
                  <stringProp name="Argument.value">satya1@gmail.com</stringProp>
                  <stringProp name="Argument.metadata">=</stringProp>
                </elementProp>
              </collectionProp>
            </elementProp>
            <stringProp name="HTTPSampler.domain">${BASE_URL_1}</stringProp>
            <stringProp name="HTTPSampler.port">9010</stringProp>
            <stringProp name="HTTPSampler.protocol">http</stringProp>
            <stringProp name="HTTPSampler.path">register-user</stringProp>
            <stringProp name="HTTPSampler.method">POST</stringProp>
            <boolProp name="HTTPSampler.follow_redirects">true</boolProp>
            <boolProp name="HTTPSampler.auto_redirects">false</boolProp>
            <boolProp name="HTTPSampler.use_keepalive">true</boolProp>
            <boolProp name="HTTPSampler.DO_MULTIPART_POST">false</boolProp>
            <boolProp name="HTTPSampler.BROWSER_COMPATIBLE_MULTIPART">false</boolProp>
            <boolProp name="HTTPSampler.image_parser">false</boolProp>
            <boolProp name="HTTPSampler.concurrentDwn">false</boolProp>
            <stringProp name="HTTPSampler.concurrentPool">6</stringProp>
            <boolProp name="HTTPSampler.md5">false</boolProp>
            <intProp name="HTTPSampler.ipSourceType">0</intProp>
          </HTTPSamplerProxy>
          <hashTree>
            <HeaderManager guiclass="HeaderPanel" testclass="HeaderManager" testname="HTTP
  Header manager" enabled="true">
              <collectionProp name="HeaderManager.headers">
                <elementProp name="Content-Type" elementType="Header">
                  <stringProp name="Header.name">Content-Type</stringProp>
                  <stringProp name="Header.value">application/x-www-form-urlencoded</stringProp>
                </elementProp>
                <elementProp name="Sec-Fetch-Site" elementType="Header">
                  <stringProp name="Header.name">Sec-Fetch-Site</stringProp>
                  <stringProp name="Header.value">same-origin</stringProp>
                </elementProp>
              </collectionProp>
            </HeaderManager>
            <hashTree/>
            <UniformRandomTimer guiclass="UniformRandomTimerGui"
  testclass="UniformRandomTimer" testname="Uniform Random Timer" enabled="true">
              <stringProp name="RandomTimer.range">16524.0</stringProp>
              <stringProp name="ConstantTimer.delay">8262</stringProp>
              <stringProp name="TestPlan.comments">Recorded time was 16524
  milliseconds</stringProp>
            </UniformRandomTimer>
            <hashTree/>
          </hashTree>
          <HTTPSamplerProxy guiclass="HttpTestSampleGui" testclass="HTTPSamplerProxy"
  testname="http://localhost:9010/cart" enabled="true">
            <boolProp name="HTTPSampler.postBodyRaw">false</boolProp>
            <elementProp name="HTTPsampler.Arguments" elementType="Arguments"
  guiclass="HTTPArgumentsPanel" testclass="Arguments" enabled="true">
              <collectionProp name="Arguments.arguments"/>
            </elementProp>
            <stringProp name="HTTPSampler.domain">${BASE_URL_1}</stringProp>
            <stringProp name="HTTPSampler.port">9010</stringProp>
            <stringProp name="HTTPSampler.protocol">http</stringProp>
            <stringProp name="HTTPSampler.path">cart</stringProp>
            <stringProp name="HTTPSampler.method">GET</stringProp>
            <boolProp name="HTTPSampler.follow_redirects">true</boolProp>
            <boolProp name="HTTPSampler.auto_redirects">false</boolProp>
            <boolProp name="HTTPSampler.use_keepalive">true</boolProp>
            <boolProp name="HTTPSampler.DO_MULTIPART_POST">false</boolProp>
            <boolProp name="HTTPSampler.BROWSER_COMPATIBLE_MULTIPART">false</boolProp>
            <boolProp name="HTTPSampler.image_parser">false</boolProp>
            <boolProp name="HTTPSampler.concurrentDwn">false</boolProp>
            <stringProp name="HTTPSampler.concurrentPool">6</stringProp>
            <boolProp name="HTTPSampler.md5">false</boolProp>
            <intProp name="HTTPSampler.ipSourceType">0</intProp>
          </HTTPSamplerProxy>
          <hashTree>
            <HeaderManager guiclass="HeaderPanel" testclass="HeaderManager" testname="HTTP
  Header manager" enabled="true">
```

```
            <collectionProp name="HeaderManager.headers">
              <elementProp name="Sec-Fetch-Site" elementType="Header">
                <stringProp name="Header.name">Sec-Fetch-Site</stringProp>
                <stringProp name="Header.value">same-origin</stringProp>
              </elementProp>
            </collectionProp>
          </HeaderManager>
          <hashTree/>
          <UniformRandomTimer guiclass="UniformRandomTimerGui"
testclass="UniformRandomTimer" testname="Uniform Random Timer" enabled="true">
            <stringProp name="RandomTimer.range">7262.0</stringProp>
            <stringProp name="ConstantTimer.delay">3631</stringProp>
            <stringProp name="TestPlan.comments">Recorded time was 7262 milliseconds</stringProp>
          </UniformRandomTimer>
          <hashTree/>
        </hashTree>
        <HTTPSamplerProxy guiclass="HttpTestSampleGui" testclass="HTTPSamplerProxy"
testname="http://localhost:9010/home" enabled="true">
          <boolProp name="HTTPSampler.postBodyRaw">false</boolProp>
          <elementProp name="HTTPsampler.Arguments" elementType="Arguments"
guiclass="HTTPArgumentsPanel" testclass="Arguments" enabled="true">
            <collectionProp name="Arguments.arguments"/>
          </elementProp>
          <stringProp name="HTTPSampler.domain">${BASE_URL_1}</stringProp>
          <stringProp name="HTTPSampler.port">9010</stringProp>
          <stringProp name="HTTPSampler.protocol">http</stringProp>
          <stringProp name="HTTPSampler.path">home</stringProp>
          <stringProp name="HTTPSampler.method">GET</stringProp>
          <boolProp name="HTTPSampler.follow_redirects">true</boolProp>
          <boolProp name="HTTPSampler.auto_redirects">false</boolProp>
          <boolProp name="HTTPSampler.use_keepalive">true</boolProp>
          <boolProp name="HTTPSampler.DO_MULTIPART_POST">false</boolProp>
          <boolProp name="HTTPSampler.BROWSER_COMPATIBLE_MULTIPART">false</boolProp>
          <boolProp name="HTTPSampler.image_parser">false</boolProp>
          <boolProp name="HTTPSampler.concurrentDwn">false</boolProp>
          <stringProp name="HTTPSampler.concurrentPool">6</stringProp>
          <boolProp name="HTTPSampler.md5">false</boolProp>
          <intProp name="HTTPSampler.ipSourceType">0</intProp>
        </HTTPSamplerProxy>
        <hashTree>
          <HeaderManager guiclass="HeaderPanel" testclass="HeaderManager" testname="HTTP Header manager" enabled="true">
            <collectionProp name="HeaderManager.headers">
              <elementProp name="Sec-Fetch-Site" elementType="Header">
                <stringProp name="Header.name">Sec-Fetch-Site</stringProp>
                <stringProp name="Header.value">same-origin</stringProp>
              </elementProp>
            </collectionProp>
          </HeaderManager>
          <hashTree/>
          <UniformRandomTimer guiclass="UniformRandomTimerGui" testclass="UniformRandomTimer" testname="Uniform Random Timer" enabled="true">
            <stringProp name="RandomTimer.range">2170.0</stringProp>
            <stringProp name="ConstantTimer.delay">1085</stringProp>
            <stringProp name="TestPlan.comments">Recorded time was 2170 milliseconds</stringProp>
          </UniformRandomTimer>
          <hashTree/>
        </hashTree>
        <HTTPSamplerProxy guiclass="HttpTestSampleGui" testclass="HTTPSamplerProxy" testname="http://localhost:9010/add-to-cart?id=101" enabled="true">
          <boolProp name="HTTPSampler.postBodyRaw">false</boolProp>
          <elementProp name="HTTPsampler.Arguments" elementType="Arguments" guiclass="HTTPArgumentsPanel" testclass="Arguments" enabled="true">
            <collectionProp name="Arguments.arguments">
```

```
              <elementProp name="id" elementType="HTTPArgument">
                <boolProp name="HTTPArgument.always_encode">false</boolProp>
                <stringProp name="Argument.name">id</stringProp>
                <stringProp name="Argument.value">101</stringProp>
                <stringProp name="Argument.metadata">=</stringProp>
              </elementProp>
            </collectionProp>
          </elementProp>
          <stringProp name="HTTPSampler.domain">${BASE_URL_1}</stringProp>
          <stringProp name="HTTPSampler.port">9010</stringProp>
          <stringProp name="HTTPSampler.protocol">http</stringProp>
          <stringProp name="HTTPSampler.path">add-to-cart</stringProp>
          <stringProp name="HTTPSampler.method">GET</stringProp>
          <boolProp name="HTTPSampler.follow_redirects">true</boolProp>
          <boolProp name="HTTPSampler.auto_redirects">false</boolProp>
          <boolProp name="HTTPSampler.use_keepalive">true</boolProp>
          <boolProp name="HTTPSampler.DO_MULTIPART_POST">false</boolProp>
          <boolProp name="HTTPSampler.BROWSER_COMPATIBLE_MULTIPART">false</boolProp>
          <boolProp name="HTTPSampler.image_parser">false</boolProp>
          <boolProp name="HTTPSampler.concurrentDwn">false</boolProp>
          <stringProp name="HTTPSampler.concurrentPool">6</stringProp>
          <boolProp name="HTTPSampler.md5">false</boolProp>
          <intProp name="HTTPSampler.ipSourceType">0</intProp>
        </HTTPSamplerProxy>
        <hashTree>
          <HeaderManager guiclass="HeaderPanel" testclass="HeaderManager" testname="HTTP Header manager" enabled="true">
            <collectionProp name="HeaderManager.headers">
              <elementProp name="Sec-Fetch-Site" elementType="Header">
                <stringProp name="Header.name">Sec-Fetch-Site</stringProp>
                <stringProp name="Header.value">same-origin</stringProp>
```

```
      </elementProp>
    </collectionProp>
  </HeaderManager>
  <hashTree/>
  <UniformRandomTimer guiclass="UniformRandomTimerGui" testclass="UniformRandomTimer" testname="Uniform Random Timer" enabled="true">
    <stringProp name="RandomTimer.range">2231.0</stringProp>
    <stringProp name="ConstantTimer.delay">1115.5</stringProp>
    <stringProp name="TestPlan.comments">Recorded time was 2231 milliseconds</stringProp>
  </UniformRandomTimer>
  <hashTree/>
</hashTree>
<HTTPSamplerProxy guiclass="HttpTestSampleGui" testclass="HTTPSamplerProxy" testname="http://localhost:9010/home" enabled="true">
  <boolProp name="HTTPSampler.postBodyRaw">false</boolProp>
  <elementProp name="HTTPsampler.Arguments" elementType="Arguments" guiclass="HTTPArgumentsPanel" testclass="Arguments" enabled="true">
    <collectionProp name="Arguments.arguments"/>
  </elementProp>
  <stringProp name="HTTPSampler.domain">${BASE_URL_1}</stringProp>
  <stringProp name="HTTPSampler.port">9010</stringProp>
  <stringProp name="HTTPSampler.protocol">http</stringProp>
  <stringProp name="HTTPSampler.path">home</stringProp>
  <stringProp name="HTTPSampler.method">GET</stringProp>
  <boolProp name="HTTPSampler.follow_redirects">true</boolProp>
  <boolProp name="HTTPSampler.auto_redirects">false</boolProp>
  <boolProp name="HTTPSampler.use_keepalive">true</boolProp>
  <boolProp name="HTTPSampler.DO_MULTIPART_POST">false</boolProp>
  <boolProp name="HTTPSampler.BROWSER_COMPATIBLE_MULTIPART">false</boolProp>
  <boolProp name="HTTPSampler.image_parser">false</boolProp>
  <boolProp name="HTTPSampler.concurrentDwn">false</boolProp>
  <stringProp name="HTTPSampler.concurrentPool">6</stringProp>
```

```
          <boolProp name="HTTPSampler.md5">false</boolProp>
          <intProp name="HTTPSampler.ipSourceType">0</intProp>
        </HTTPSamplerProxy>
        <hashTree>
          <HeaderManager guiclass="HeaderPanel" testclass="HeaderManager" testname="HTTP Header manager" enabled="true">
            <collectionProp name="HeaderManager.headers">
              <elementProp name="Sec-Fetch-Site" elementType="Header">
                <stringProp name="Header.name">Sec-Fetch-Site</stringProp>
                <stringProp name="Header.value">same-origin</stringProp>
              </elementProp>
            </collectionProp>
          </HeaderManager>
          <hashTree/>
          <UniformRandomTimer guiclass="UniformRandomTimerGui" testclass="UniformRandomTimer" testname="Uniform Random Timer" enabled="true">
            <stringProp name="RandomTimer.range">1216.0</stringProp>
            <stringProp name="ConstantTimer.delay">608</stringProp>
            <stringProp name="TestPlan.comments">Recorded time was 1216 milliseconds</stringProp>
          </UniformRandomTimer>
          <hashTree/>
        </hashTree>
        <HTTPSamplerProxy guiclass="HttpTestSampleGui" testclass="HTTPSamplerProxy" testname="http://localhost:9010/cart" enabled="true">
          <boolProp name="HTTPSampler.postBodyRaw">false</boolProp>
          <elementProp name="HTTPsampler.Arguments" elementType="Arguments" guiclass="HTTPArgumentsPanel" testclass="Arguments" enabled="true">
            <collectionProp name="Arguments.arguments"/>
          </elementProp>
          <stringProp name="HTTPSampler.domain">${BASE_URL_1}</stringProp>
          <stringProp name="HTTPSampler.port">9010</stringProp>
          <stringProp name="HTTPSampler.protocol">http</stringProp>
          <stringProp name="HTTPSampler.path">cart</stringProp>
          <stringProp name="HTTPSampler.method">GET</stringProp>
          <boolProp name="HTTPSampler.follow_redirects">true</boolProp>
          <boolProp name="HTTPSampler.auto_redirects">false</boolProp>
          <boolProp name="HTTPSampler.use_keepalive">true</boolProp>
          <boolProp name="HTTPSampler.DO_MULTIPART_POST">false</boolProp>
          <boolProp name="HTTPSampler.BROWSER_COMPATIBLE_MULTIPART">false</boolProp>
          <boolProp name="HTTPSampler.image_parser">false</boolProp>
          <boolProp name="HTTPSampler.concurrentDwn">false</boolProp>
          <stringProp name="HTTPSampler.concurrentPool">6</stringProp>
          <boolProp name="HTTPSampler.md5">false</boolProp>
          <intProp name="HTTPSampler.ipSourceType">0</intProp>
        </HTTPSamplerProxy>
        <hashTree>
          <HeaderManager guiclass="HeaderPanel" testclass="HeaderManager" testname="HTTP Header manager" enabled="true">
            <collectionProp name="HeaderManager.headers">
              <elementProp name="Sec-Fetch-Site" elementType="Header">
                <stringProp name="Header.name">Sec-Fetch-Site</stringProp>
                <stringProp name="Header.value">same-origin</stringProp>
              </elementProp>
            </collectionProp>
          </HeaderManager>
          <hashTree/>
          <UniformRandomTimer guiclass="UniformRandomTimerGui" testclass="UniformRandomTimer" testname="Uniform Random Timer" enabled="true">
            <stringProp name="RandomTimer.range">1172.0</stringProp>
            <stringProp name="ConstantTimer.delay">586</stringProp>
            <stringProp name="TestPlan.comments">Recorded time was 1172 milliseconds</stringProp>
          </UniformRandomTimer>
          <hashTree/>
        </hashTree>
```

```
<HTTPSamplerProxy guiclass="HttpTestSampleGui" testclass="HTTPSamplerProxy" testname="http://localhost:9010/place-order" enabled="true">
  <boolProp name="HTTPSampler.postBodyRaw">false</boolProp>
  <elementProp name="HTTPsampler.Arguments" elementType="Arguments" guiclass="HTTPArgumentsPanel" testclass="Arguments" enabled="true">
    <collectionProp name="Arguments.arguments"/>
  </elementProp>
  <stringProp name="HTTPSampler.domain">${BASE_URL_1}</stringProp>
  <stringProp name="HTTPSampler.port">9010</stringProp>
  <stringProp name="HTTPSampler.protocol">http</stringProp>
  <stringProp name="HTTPSampler.path">place-order</stringProp>
  <stringProp name="HTTPSampler.method">GET</stringProp>
  <boolProp name="HTTPSampler.follow_redirects">true</boolProp>
  <boolProp name="HTTPSampler.auto_redirects">false</boolProp>
  <boolProp name="HTTPSampler.use_keepalive">true</boolProp>
  <boolProp name="HTTPSampler.DO_MULTIPART_POST">false</boolProp>
  <boolProp name="HTTPSampler.BROWSER_COMPATIBLE_MULTIPART">false</boolProp>
  <boolProp name="HTTPSampler.image_parser">false</boolProp>
  <boolProp name="HTTPSampler.concurrentDwn">false</boolProp>
  <stringProp name="HTTPSampler.concurrentPool">6</stringProp>
  <boolProp name="HTTPSampler.md5">false</boolProp>
  <intProp name="HTTPSampler.ipSourceType">0</intProp>
</HTTPSamplerProxy>
<hashTree>
  <HeaderManager guiclass="HeaderPanel" testclass="HeaderManager" testname="HTTP Header manager" enabled="true">
    <collectionProp name="HeaderManager.headers">
      <elementProp name="Sec-Fetch-Site" elementType="Header">
        <stringProp name="Header.name">Sec-Fetch-Site</stringProp>
        <stringProp name="Header.value">same-origin</stringProp>
      </elementProp>
    </collectionProp>
```

```
            </HeaderManager>
            <hashTree/>
            <UniformRandomTimer guiclass="UniformRandomTimerGui"
testclass="UniformRandomTimer" testname="Uniform Random Timer" enabled="true">
              <stringProp name="RandomTimer.range">1861.0</stringProp>
              <stringProp name="ConstantTimer.delay">930.5</stringProp>
              <stringProp name="TestPlan.comments">Recorded time was 1861 milliseconds</stringProp>
            </UniformRandomTimer>
            <hashTree/>
          </hashTree>
        </hashTree>
        <ResultCollector guiclass="ViewResultsFullVisualizer" testclass="ResultCollector"
testname="View Results Tree" enabled="true">
          <boolProp name="ResultCollector.error_logging">false</boolProp>
          <objProp>
            <name>saveConfig</name>
            <value class="SampleSaveConfiguration">
              <time>true</time>
              <latency>true</latency>
              <timestamp>true</timestamp>
              <success>true</success>
              <label>true</label>
              <code>true</code>
              <message>true</message>
              <threadName>true</threadName>
              <dataType>true</dataType>
              <encoding>false</encoding>
              <assertions>true</assertions>
              <subresults>true</subresults>
              <responseData>false</responseData>
              <samplerData>false</samplerData>
              <xml>false</xml>
              <fieldNames>true</fieldNames>
              <responseHeaders>false</responseHeaders>
              <requestHeaders>false</requestHeaders>
              <responseDataOnError>false</responseDataOnError>
              <saveAssertionResultsFailureMessage>true</saveAssertionResultsFailureMessage>
              <assertionsResultsToSave>0</assertionsResultsToSave>
              <bytes>true</bytes>
              <sentBytes>true</sentBytes>
              <url>true</url>
              <threadCounts>true</threadCounts>
              <idleTime>true</idleTime>
              <connectTime>true</connectTime>
            </value>
          </objProp>
          <stringProp name="filename"></stringProp>
        </ResultCollector>
        <hashTree/>
      </hashTree>
    </hashTree>
  </hashTree>
</jmeterTestPlan>
```