KLE Society's
KLE Technological University

Software Testing

On

Raita Samparka Kendra

Bachelor of Engineering

In

Computer Science and Engineering

Submitted By

Team No: 05

PRADEEP CHEGUR	**01FE19BCS294**
PRAVEEN TAKKANNAVAR	**01FE19BCS298**
SWAGAT INGALAGAON	**01FE19BCS299**
NAVEEN DODDAMANI	**01FE19BCS305**

Faculty In charge
Ms. Kavitha H S

SCHOOL OF COMPUTER SCIENCE & ENGINEERING

HUBLI–580 031 (India).

Academic year 2021-22

PROBLEM STATEMENT

Raita Sampark Kendra:

- Farmers will buy the fertilizers at RSK, in order to purchase they have to registered by the executives.
- Farmers should complete the KYC of his/her farm (register the farm details, property id, area etc.), a farmer can register one or more farm land.
- Executives will make an entry of each purchase made by the farmer along with the crop details.
- Maintain the history of the crops in their farm which will help them to claim insurance in case of disaster.

PART A: Software Requirements specifications

Functional Requirements

1. Executives shall be able to register the farmer.
2. Executives shall be able to login to the website.
3. Both Executives and Farmers shall be able to view the details of fertilizers.
4. Farmer shall be able to Buy/Add fertilizers.
5. Executives shall be able to make an entry of each purchase made by the farmer along with the crop details.
6. Executives shall be able to maintain the history of the crops in the farm.
7. Farmers shall be able to view contacts of Executives.

Non-Functional requirements

1. The RSK system should be available to all the farmers during normal working hours (Mon-Fri, 10 am to 6 pm).
2. Downtime within normal working hours should not exceed 5 seconds in any one day.
3. The system latency should not exceed 3000ms for one process.
4. The system should not assign more than 1500000bytes of memory for one process.
5. Farmers of the RSK system should identify themselves using their farmer id card.

Software Architecture

Layered Architecture: RSK System

User Interfaces

Web Browser	**RSK App**

Configuration Services

Login	**Application Management**	**Form and Menu Manager**	**Identity Management**

Application Services

Security Management	**Farmer Information Management**
Data Import and Export	

Utility Management

Transaction Management	**Farmer Database**

Explanation:

1. The top layer is responsible for implementing the user interface. In this case, the UI has been implemented using a web browser.
2. The second layer provides the user interface functionality that is delivered through the web browser. It includes components to allow farmers to log in to the system and checking components that ensure that the operations they use are allowed by their role. This layer includes form and menu management components that present information to farmers.
3. The third layer implements the functionality of the system and provides components that implement system security, Farmer information creation and updating, import and export of Farmer data databases.
4. Finally, the lowest layer, which is built using a commercial database management system, provides transaction management and persistent data storage.

Name	Layered architecture
Description	Organizes the system into layers with related functionality associated with each layer. A layer provides services to the layer above it so the lowest-level layers represent core services that are likely to be used throughout the system.
Example	A layered model of a system for farmer data base.
When used	Used when building new facilities on top of existing systems; when the development is spread across several teams with each team responsibility for a layer of functionality; when there is a requirement for multi-level security
Advantages	Allows replacement of entire layers so long as the interface is maintained. Redundant facilities (e.g., authentication) can be provided in each layer to increase the dependability of the system
Disadvantages	In practice, providing a clean separation between layers is often difficult and a high-level layer may have to interact directly with lower-level layers rather than through the layer immediately below it. Performance can be a problem because of multiple levels of interpretation of a service request as it is processed at each layer.

PART B: Test plan and Test cases

Requirement id	Test id	Input	Expected output	Test status
1	**1.1**	**About us**	**Display details of the RSK System.**	**PASS**
	1.2	**Know more**	**Takes to official website of RSK**	**PASS**
	1.3	**Contact Us**	**Display mail and mobile numbers**	**PASS**
	1.4	**Return to home**	**Takes back to Home Page**	**PASS**
2	**2.1**	**Schemes**	**Display different types of schemes to apply**	**PASS**
	2.2	**Click here to apply**	**Direct links to available schemes**	**PASS**
3	**3.1**	**Executive Login**	**Display new page for executive login**	**PASS**
	3.2	**Login Button**	**Takes to Registration form Of farmers**	**PASS**
	3.3	**Reset**	**Reset the details filled in the form**	**PASS**
	3.4	**Submit**	**Form submitted successfully**	**PASS**
	3.5	**Checkbox**	**Add total amount of fertilizers**	**PASS**
4	**4.1**	**Executive logout**	**Log out from executive mode**	**PASS**
	4.2	**Return Home**	**Return to home page**	**PASS**

Testcase 0

Title: Contact Details
Description: A user should be able to view all the contact details available for purchase of fertilizers.
Precondition: Home page consisting of Contact details button.
Assumption: The full stack website is running.
Test Steps:

1. The home page must be displayed.
2. Clicking on about us button.

Expected Result: User is able to view the contact list with mail ids.

Testcase 1

Title: Available Schemes
Description: A user should be able to view all the available schemes for purchase of fertilizers.
Precondition: Home page consisting of Schemes button.
Assumption: The full stack website is running.
Test Steps:

1. The home page must be displayed.
2. Clicking on schemes button.

Expected Result: User is able to view the available schemes.

Testcase 2

Title: Login for Executives
Description: Executive can login through login portal by entering username and password.
Precondition: Executive page consisting of Login button.
Assumption: The full stack website is running.
Test Steps:

1. The executive page must be displayed.
2. Entering valid username and password.
3. Login through the portal.

Expected Result: Executive should make entry of farmers through form.

PART C:

Description of Testing Tool:

Apache JMeter is an open source, Java-based, load testing tool that can be used to analyze the functional behavior of a system and measure the performance of a system under a load test. A load test will simulate end-user behavior that approach the limits of an application's specifications. Apache JMeter can be used to simulate varying or heavy loads on singular or multiple servers, networks or objects to test a system's strength.

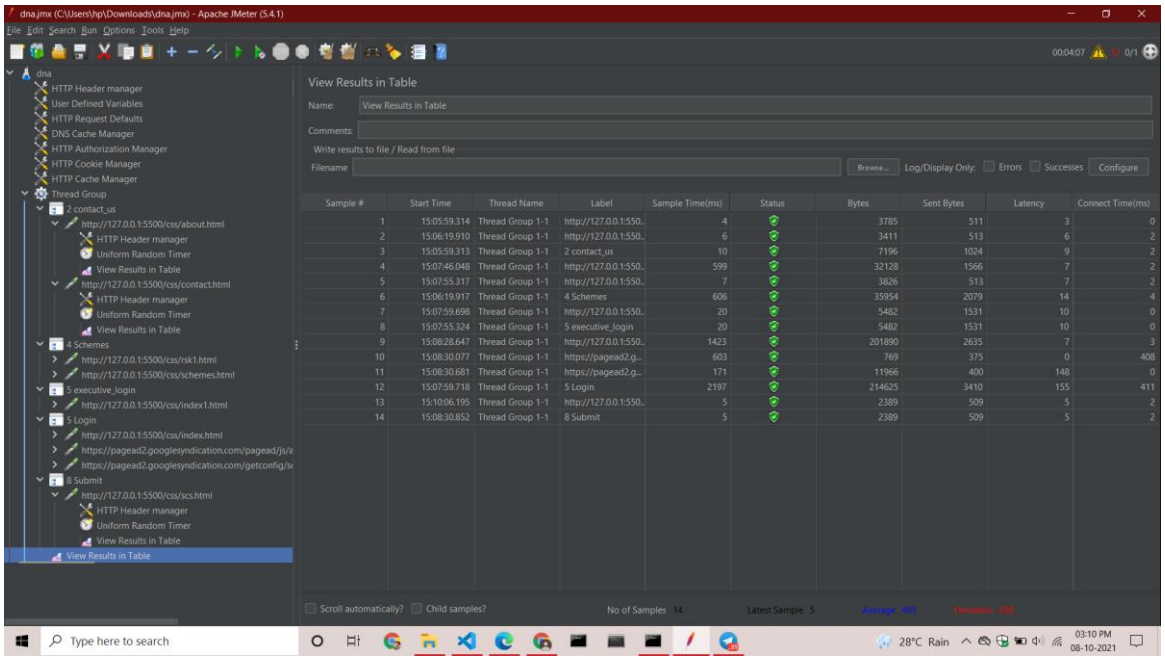
Jmeter works by simulating groups of users that send requests to a server or network, then returning statistics back to a user through visual diagrams. Apache Jmeter's GUI looks similar to a browser; however, Jmeter cannot render HTML pages or the JavaScript found in the pages as a browser would. Additionally, Jmeter supports the use of plug-ins. Plug-ins supported by JMeter will extend the functionality of Jmeter and can be installed through the Plugin Manager.

BENEFITS OF APACHE JMETER:

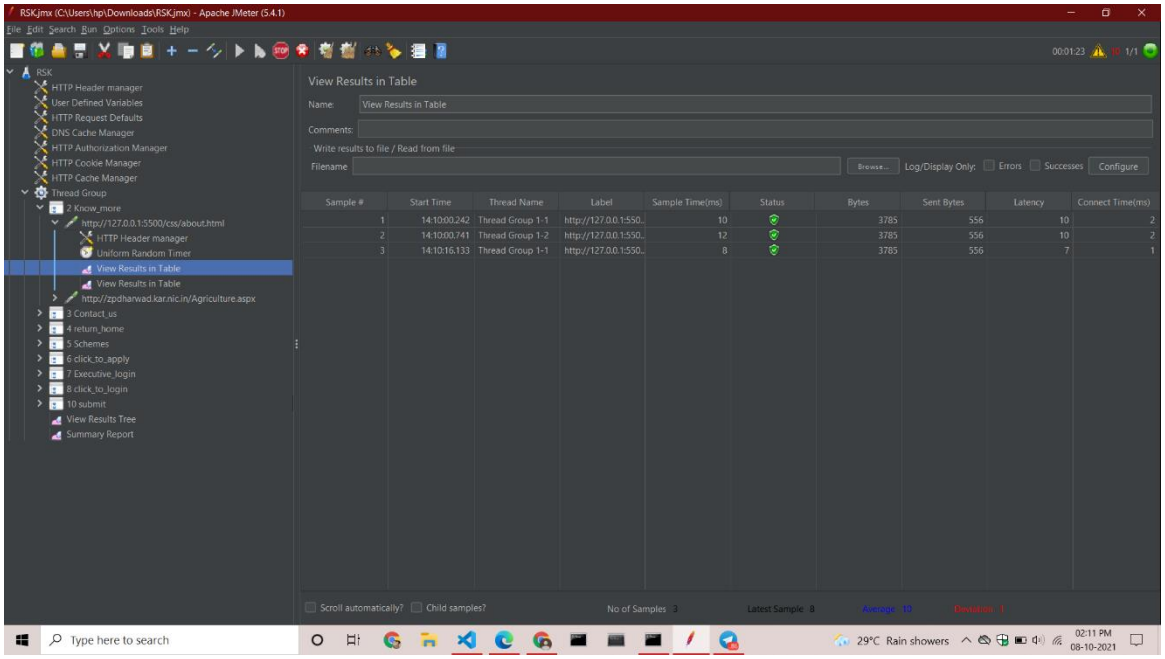
- Open source code base.
- Support for browser plug-ins.
- Support for offline analysis of test results.
- Displays test results in a variety of ways, including charts, trees, tables or log files.
- User-friendly graphical user interface (GUI).
- Support for multiple types of tests and basic protocols.

Screenshots:

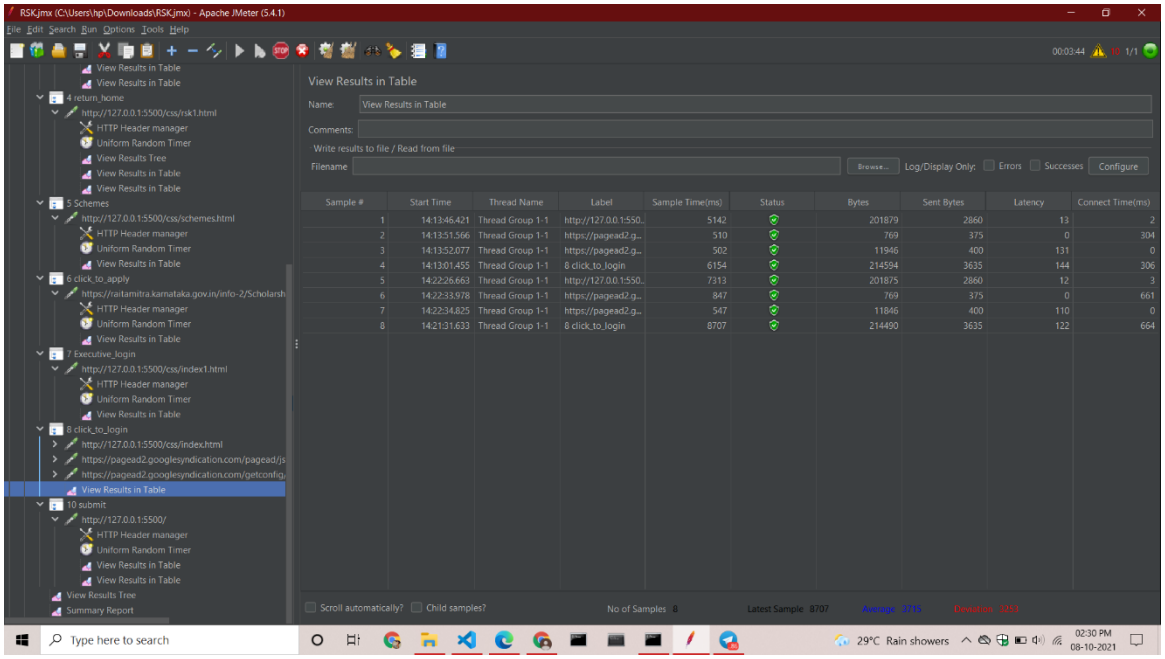
Overall Process:



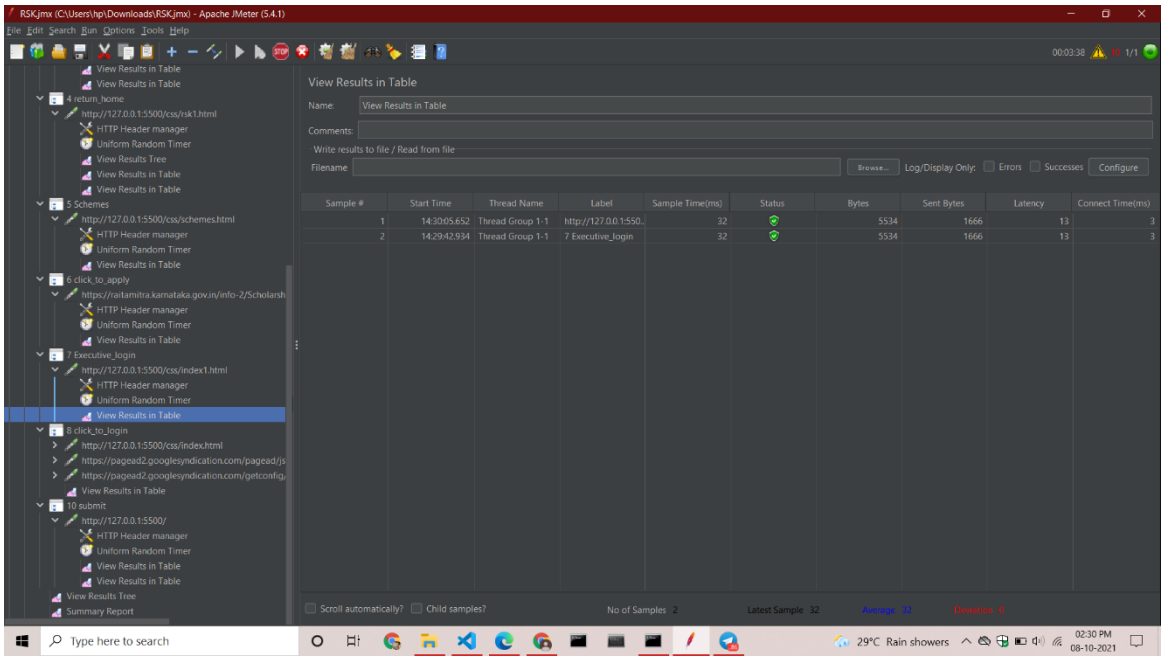
1.2 Know More



3.2 login button



3.1 Executive login



1.4 Return To Home