

Non - Functional Testing

Group - 5

1. Security

1) CSRF (Cross Site Request Forgery) Protection

The site uses CSRF tokens for all forms and POST requests, ensuring that malicious actors cannot forge requests on behalf of authenticated users.

CSRF Middleware:

```
46 MIDDLEWARE = [  
47     'django.middleware.security.SecurityMiddleware',  
48     'django.contrib.sessions.middleware.SessionMiddleware',  
49     'django.middleware.common.CommonMiddleware',  
50     'django.middleware.csrf.CsrfViewMiddleware',  
51     'django.contrib.auth.middleware.AuthenticationMiddleware',  
52     'django.contrib.messages.middleware.MessageMiddleware',  
53     'django.middleware.clickjacking.XFrameOptionsMiddleware',  
54 ]
```

CSRF Tokens in forms:

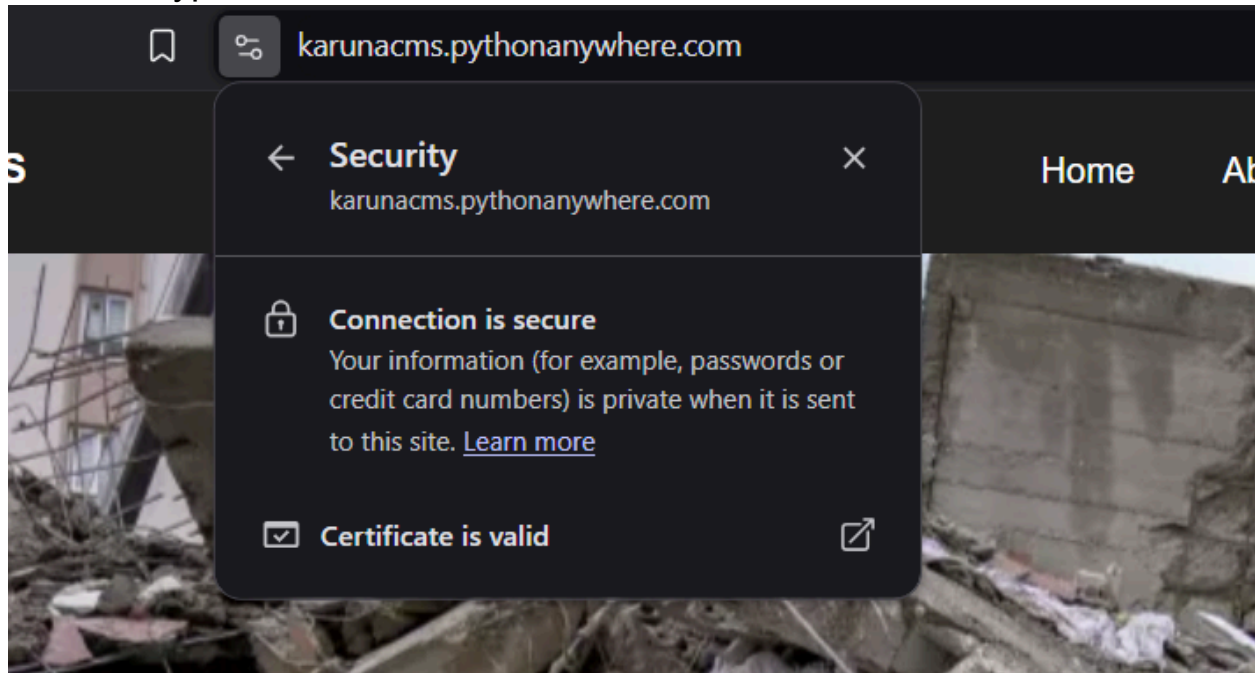
```
56 <div class="report-incident">  
57     <h1>Report an Incident</h1>  
58     <form method="POST">  
59         {% csrf_token %}  
60         <label for="incident-type">Incident  
61         <input type="text" name="name" id="i  
62  
63         <label for="description">Description
```

2) Authenticate System

The site implements Django's secure authentication framework, ensuring encrypted password storage using PBKDF2 algorithm with SHA256 hash. (User passwords are not visible to admin, thereby maintaining data integrity).

3) HTTPS with Secure Headers

The deployment on PythonAnywhere supports HTTPS, ensuring encrypted communication between users and the server.



4) Permissions

Access to sensitive data and operations is restricted based on user roles and permissions using Django's authentication and authorization system. The user is always prompted for location permission while visiting the site.

5) Database Security

Django ORM prevents direct database access and mitigates risks like SQL injection. Only validated queries can interact with the database.

2. Load Testing of Karuna CMS

We are using **Apache - JMeter v5.6.3** with our site deployed on the pythonanywhere server.

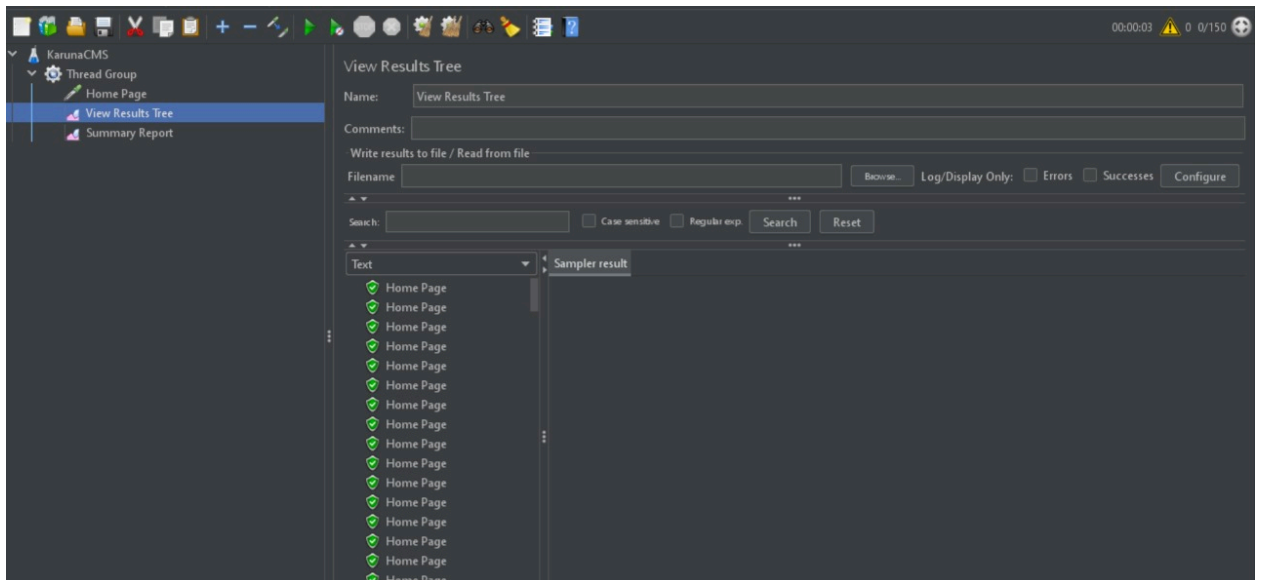
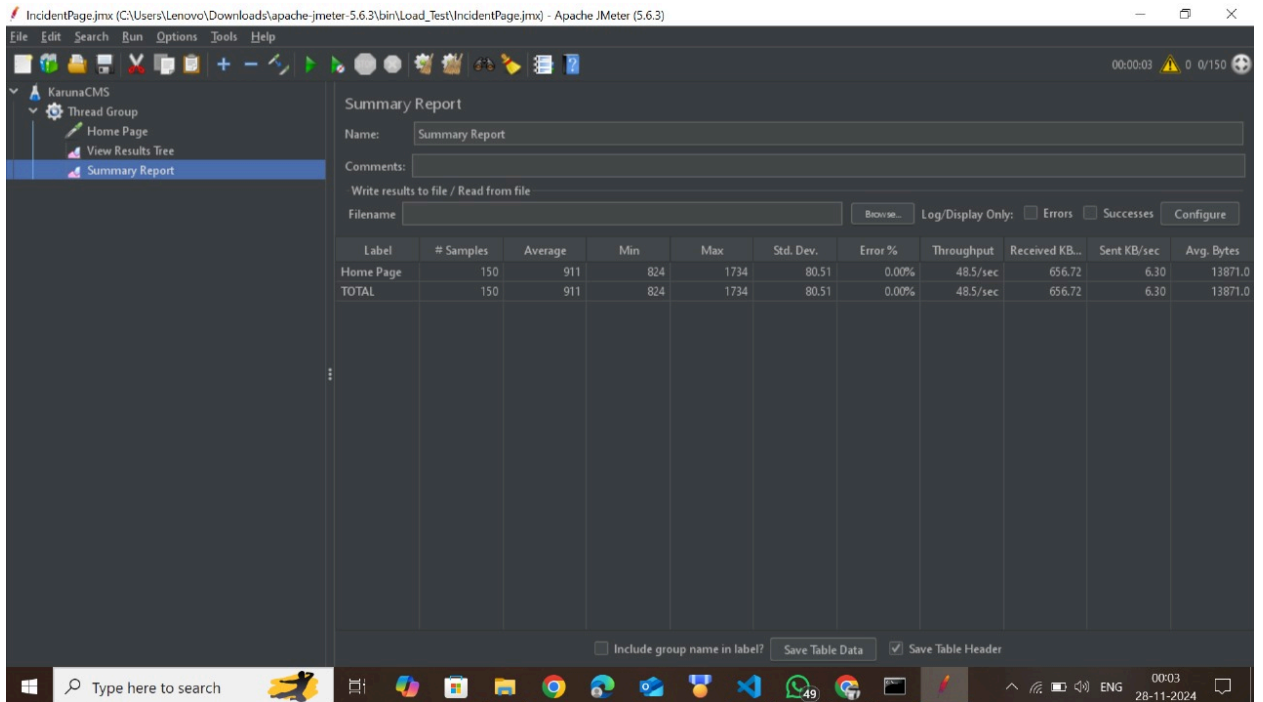
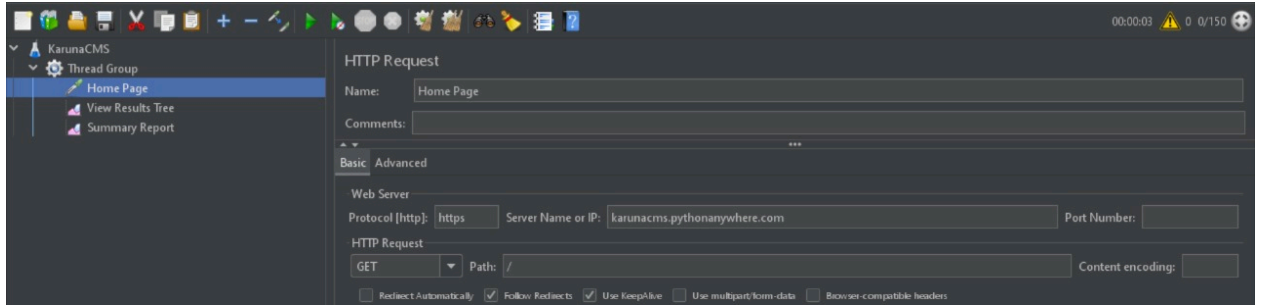
For Load Testing of our Site we are using same parameters:

No. of Users : 150

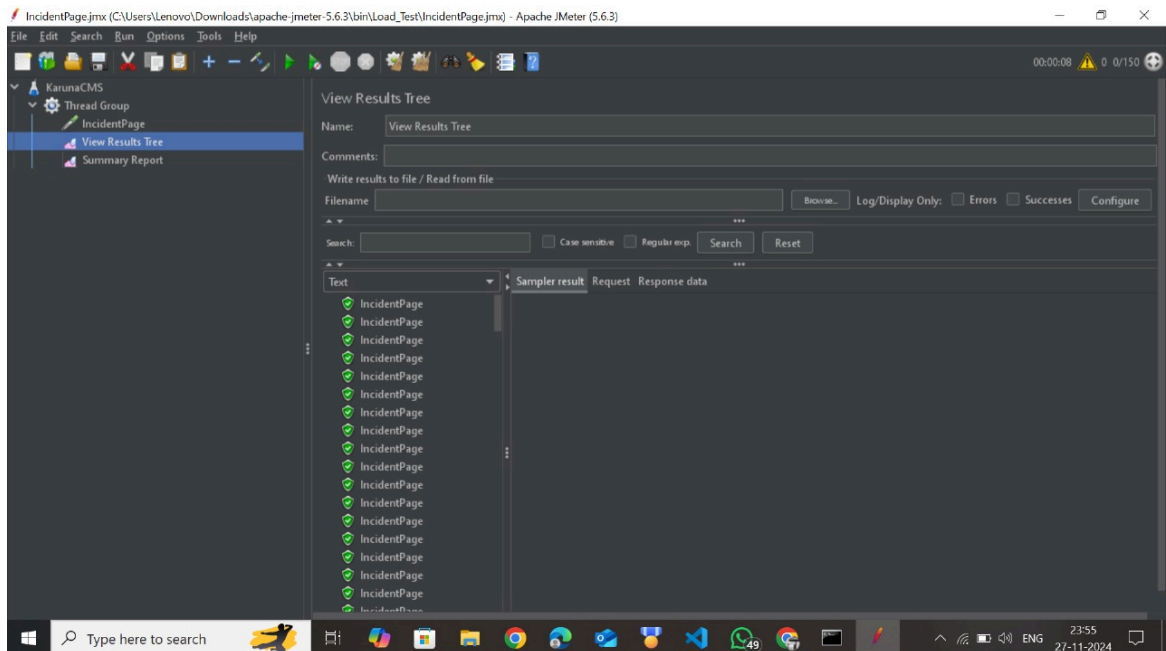
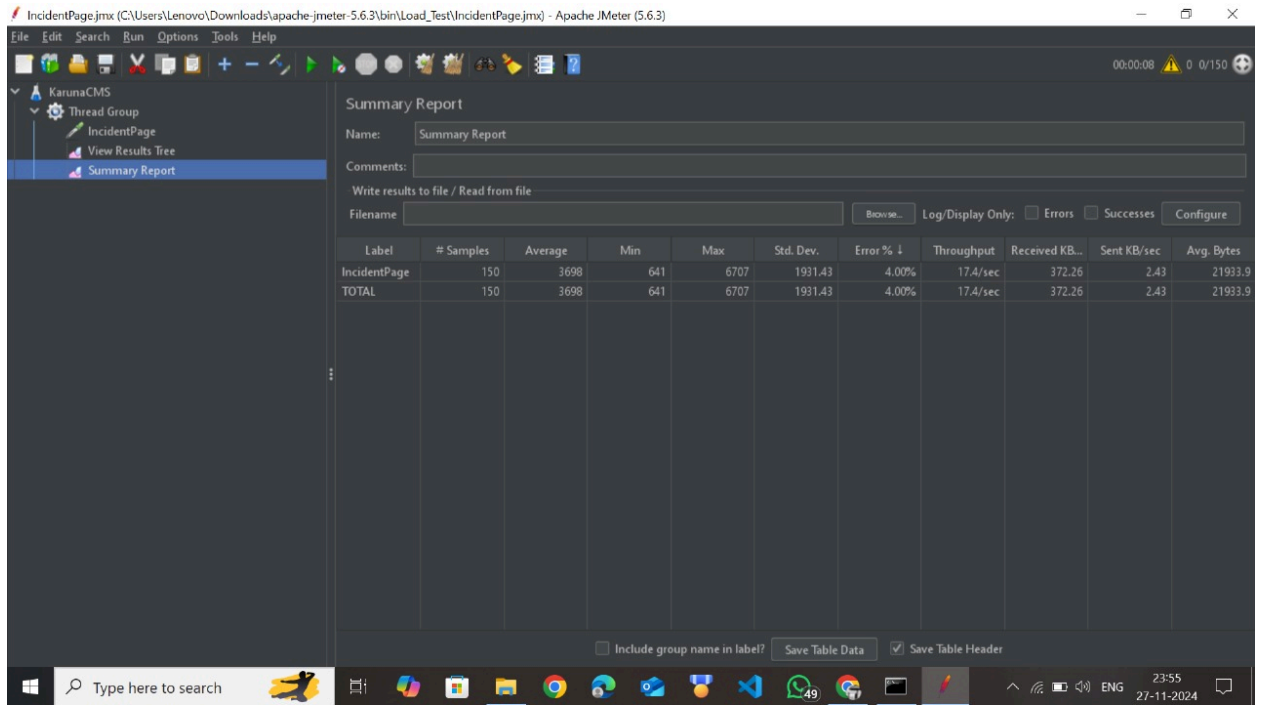
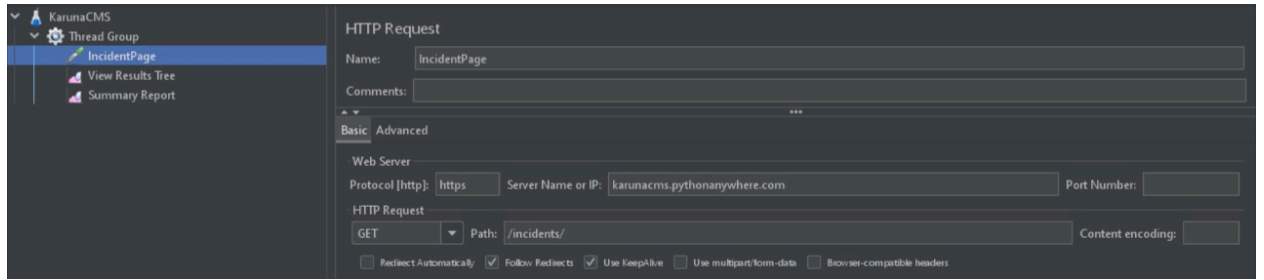
Ramp-Up Period: 2 sec.

Loop Count : 1

1) Home Page



2) Incident Page



3) Guidelines Page

The screenshot displays the Apache JMeter 5.6.3 interface. The left sidebar shows the Test Plan structure with 'Guidelines' selected. The main panel shows the 'HTTP Request' configuration for a test named 'Guidelines'. The configuration includes a Protocol of 'https', a Server Name or IP of 'karunacms.pythonanywhere.com', and a Path of '/guidelines/'. The 'Basic' tab is active, showing options like 'Redirect Automatically', 'Follow Redirects', 'Use Keep Alive', 'Use multipart/form-data', and 'Browser-compatible headers'.

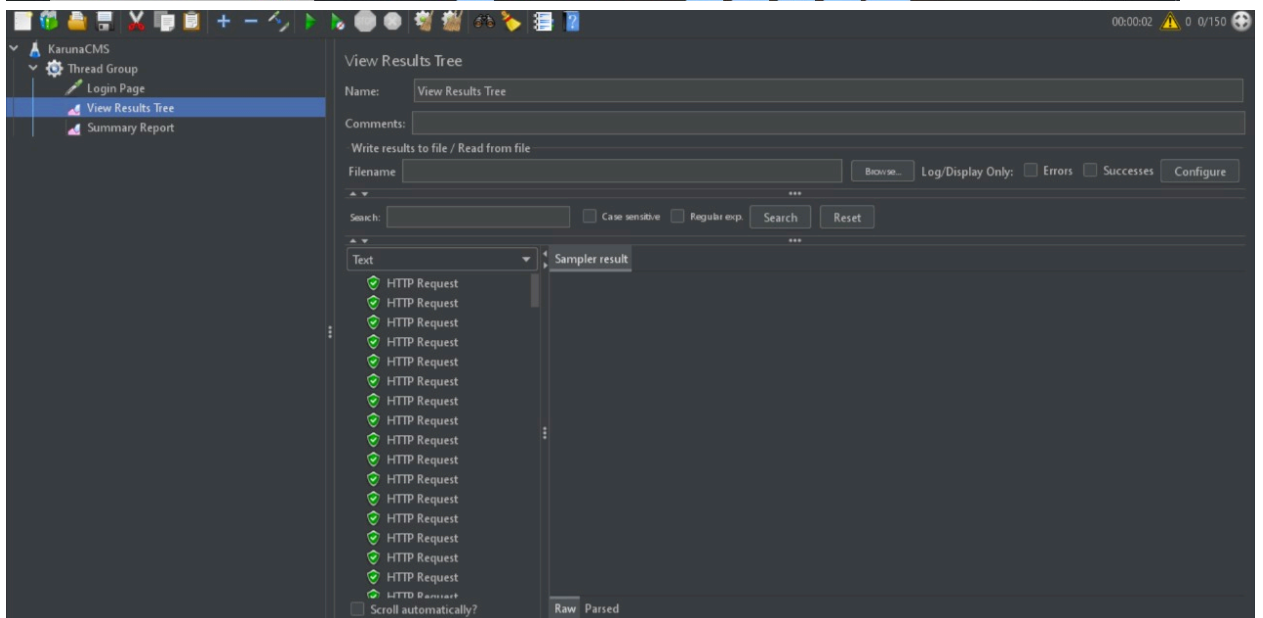
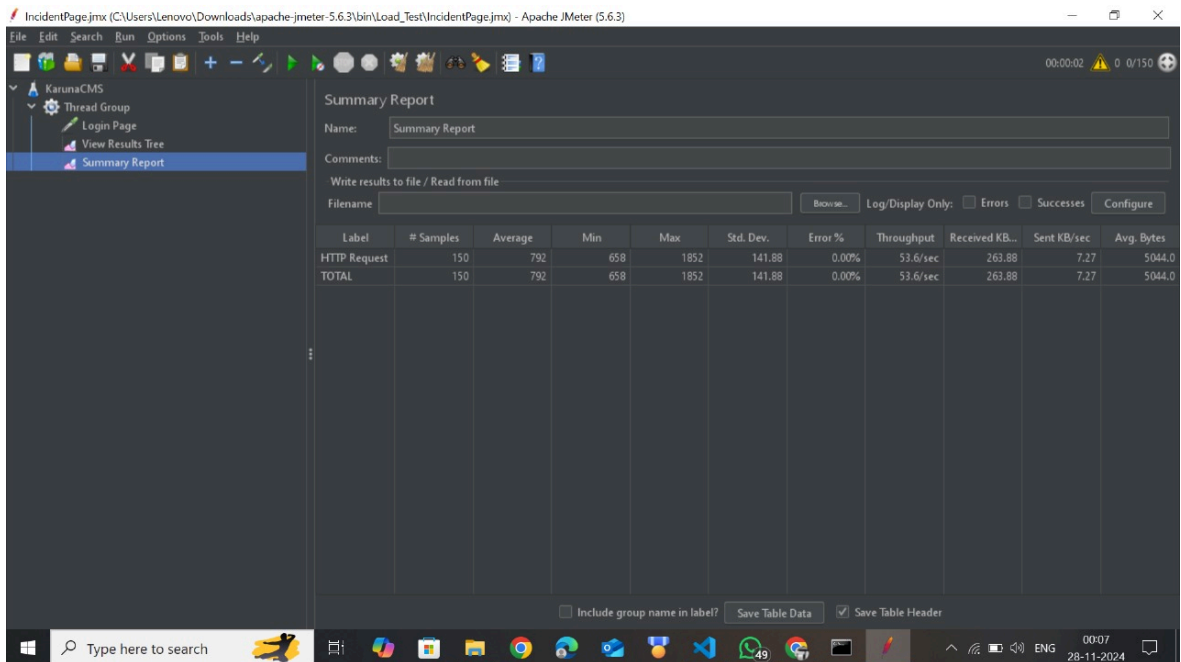
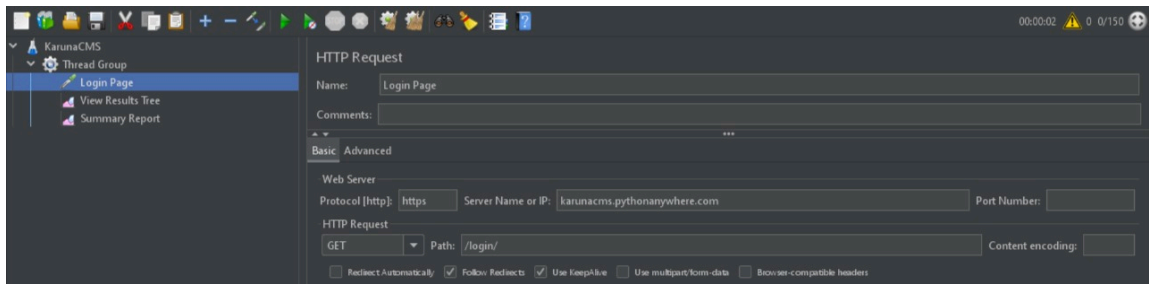
Below the HTTP Request configuration, the 'Summary Report' is displayed. It shows a table with the following data:

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
Resources Page	150	983	641	2419	385.99	0.00%	39.8/sec	345.31	5.48	8880.0
TOTAL	150	983	641	2419	385.99	0.00%	39.8/sec	345.31	5.48	8880.0

The 'Summary Report' also includes a 'Write results to file / Read from file' section with a 'Filename' field and a 'Browse...' button. The 'Log/Display Only' section has checkboxes for 'Errors' and 'Successes'. The 'Include group name in label?' checkbox is checked, and the 'Save Table Data' checkbox is also checked.

Below the Summary Report, the 'View Results Tree' is displayed. It shows a list of test results for 'Resources Page' with a 'Sampler result' column. The 'View Results Tree' also includes a 'Write results to file / Read from file' section with a 'Filename' field and a 'Browse...' button. The 'Log/Display Only' section has checkboxes for 'Errors' and 'Successes'. The 'Include group name in label?' checkbox is checked, and the 'Save Table Data' checkbox is also checked.

4) Login Page



5) Register Page

