Journal of Engineering Sciences and Information Technology
Volume (6), Issue (5) : 30 Sep 2022
P: 51 - 80

ISSN: 2522-3321

مجلة العلوم الهندسية وتكنولوجيا المعلومات
المجلد (6)، العدد (5) : 30 سبتمبر 2022م
ص: 51 - 80

Framework for testing the performance of web applications

Anwar Habeeb Eisa
Syrian Virtual University || Syria
Mohamed Ali Mohamed
Faculty of Informatics & Communications Engineering || Arab International University || Syria

Abstract: The importance of software testing, particularly performance testing for web applications, is growing with the widespread use of e-commerce applications, where there are many commercial and free tools for conducting performance testing, verifying application service quality standards, and each tool has different features that make it meet certain stages of testing, so it was necessary to integrate these tools in order to reach a comprehensive framework that meets all stages and types of required tests.

This research has built a framework that meets all stages and requirements of the testing process by identifying all types of performance tests required, systematically searching for free tools capable of implementing them, and proposing methodologies for work at each stage of the test.

The proposed framework PrefTester consists of a tool for managing the testing process PrefTester Manager and two testing tools, GTmetrix & JMeter, in order to benefit from the features of each of them at a testing stage. The test results are more accurate and comprehensive, which will serve as a reference for application developers to study and make possible improvements in order to achieve user satisfaction, reach higher corporate productivity, and achieve more profits.

Keywords: Testing- Web applications- Performance- Response time.

إطار عمل لاختبار أداء تطبيقات الويب

أنور حبيب عيسى
الجامعة الافتراضية السورية || سوريا
محمد علي محمد
كلية الهندسة المعلوماتية والاتصالات || الجامعة العربية الدولية || سوريا

المستخلص: تتزايد أهمية اختبارات البرمجيات ولاسيما اختبارات الأداء لتطبيقات الويب مع الانتشار الواسع لتطبيقات التجارة الالكترونية، حيث توجد العديد من الأدوات التجارية والمجانية من أجل إجراء عملية اختبار الأداء، والتحقق من معايير جودة خدمة التطبيقات، وتمتلك كل أداة ميزات مختلفة عن الأخرى تجعلها تلبي مراحل معينة من مراحل الاختبار، لذلك كان من الضروري إجراء تكامل بين هذه الأدوات بهدف الوصول إلى إطار شامل يلبي كافة مراحل وأنواع الاختبارات المطلوبة.

تم في هذا البحث بناء إطار عمل يلبي كافة مراحل ومتطلبات عملية الاختبار من خلال التعرف على كافة أنواع اختبارات الأداء المطلوبة، والبحث المنهج عن الأدوات المجانية القادرة على تنفيذها، واقتراح منهجيات للعمل في كل مرحلة من مراحل الاختبار.

يتكون الإطار المقترح PrefTester من أداة لإدارة عملية الاختبار PrefTester Manager وأداتين للاختبار هما GTmetrix & JMeter بهدف الاستفادة من ميزات كل منها في مرحلة من مراحل الاختبار، وهذا ما جعل من بيئة الاختبار أقرب ما يمكن إلى السيناريوهات التي قد

يتعرض لها المستخدم في البيئة الواقعية، وأصبحت نتائج الاختبار أكثر دقة وشمولية، والتي ستشكل مرجعا لمطوري التطبيقات بهدف دراستها وإجراء التحسينات الممكنة من أجل تحقيق رضا المستخدم، والوصول إلى إنتاجية أعلى للشركات، وتحقيق أرباح أكثر.

الكلمات المفتاحية: الاختبار، تطبيقات الويب، الأداء، زمن الاستجابة.

1.المقدمة.

شهد العقد الأخير زيادة كبيرة في عدد مستخدمي تطبيقات الويب في مختلف قطاعات الأعمال وهذا تطلب الاهتمام بالكثير من المتطلبات غير الوظيفية، ومع توسّع نطاق تطبيقات الويب و ازدياد تعقيدها، حيث يصل عدد المستخدمين في بعض الأوقات إلى عشرات الملايين أو حتى مئات الملايين كمواقع الخدمات الحكومية أو وسائل التواصل الاجتماعي، حيث يؤدي النمو السريع لحركة المرور على مواقع الويب إلى بطء الاستجابة بشكل متكرر بسبب الحمل الزائد، و من الممكن أن تصل عيوب أداء الموقع إلى مستوى غير مقبول، سيتسبب ذلك في إنهاء المستخدمين لمعاملاتهم على الموقع قبل الأوان، وهذا سيؤدي إلى انخفاض العائدات والأرباح المتوقعة لهذه المواقع.

يختلف اختبار الأداء عن الاختبار الوظيفي، في اختبار الأداء، نركز بشكل أساسي على السرعة، بينما يتعلق الاختبار الوظيفي بصحة سلوك التطبيق، على الرغم من أن اختبارات الأداء هي جزء من دورة الاختبارات المستمرة إلى جانب الاختبارات الوظيفية، إلا أنه يجب تفسير نتائج اختبار الأداء فقط بعد اجتياز التطبيق للاختبارات الوظيفية.

عندما يصبح النظام مستقرًا وظيفيًا، يتم بدء اختبار الأداء لتحديد مدى سرعة أداء بعض جوانب تطبيق الويب في ظل عبء عمل معين، وبعبارة أخرى، فإن فحص الوظائف في ظروف الحمل والضغط العالي يسمى اختبار الأداء.

تم في هذا البحث استعراض المفاهيم والقضايا النظرية المتعلقة باختبارات الأداء، ثم الانتقال إلى دراسة الأبحاث ذات الصلة، والتي تناولت عدة منهجيات لإجراء اختبارات كانت معتمدة على ميزات أداة الاختبار المستخدمة فقط، وبالتالي كانت تغطي مراحل معينة من مراحل الأداء، وأنواع محددة من الاختبارات، وعدم مراعاة أن يكون المستخدم المولد من أداة الاختبار نموذجا مشابها تماما لما يقوم به المستخدم الحقيقي في سلوكه على الموقع أثناء مدة الاختبار، لذلك تم اقتراح تصميم أداة PrefTester Manger بواسطة إطار العمل Laravel من أجل تحقيق مراحل الاختبار التي لا يمكن أن تؤديها أداة الاختبار، حيث أن وظيفة أداة الاختبار في تنفيذ خطة الاختبار فقط، وتم اختيار أداتي الاختبارJMeter & GTmetrix لأنهما معا قادران على تلبية متطلبات الخطة واكتشاف ثغرات أكثر في أداء التطبيق، وبالتالي أصبحت نتائج الاختبار أكثر دقة وأكثر شمولية، وتم التخلص من العيوب الموجودة في المنهجيات السابقة.

1.1. مشكلة البحث:

يمثل اختبار الأداء أحد أهم أنواع الاختبارات غير الوظيفية لمعرفة كيفية استخدام الموارد بشكل أمثل لضمان أفضل زمن استجابة للمستخدم، ويتطلب دراسة الزمن الذي يجب أن ينتظره المستخدم قبل عرض صفحة معينة، والزمن الذي يجب أن ينتظره المستخدم خلالها تنفيذ عملية ما، عادة ما يكون زمن الاستجابة هو مصدر قلق للمستخدم، والانتاجية مصدر قلق في الشركات، يساعد اختبار الأداء في إعطاء معلومات دقيقة عن المشاكل والاختناقات التي قد تحصل مستقبلا، والتي قد تؤثر على سمعة الشركات والمساعدة في اتخاذ قرارات تتعلق بتوسع التطبيقات، واستقرار عمل هذه التطبيقات بزيادة عدد المستخدمين.

بعد مراجعة الأدبيات السابقة (Srivastava, 2021) (Hayek, 2019)اتضح لدينا وجود منهجيتان أساسيتان لإتمام عملية اختبار تطبيقات الويب ولكل منها مزايا وعيوب، سنقوم باستعراضهما فيما يلي وهما

1- Manual testing (الاختبار اليدوي): تركز هذه المنهجية على فهم متطلبات التطبيق والوظائف التي يؤديها، وتحديد حالات الاختبار الممكنة، تستغرق هذه الطريقة وقتًا طويلاً وتحتاج إلى قدر كبير من الموارد البشرية لأداء المهام المطلوبة وهذا يساعد في كشف بعض العيوب الأولية، ونظرًا لأن هذه الطريقة يقوم بها البشر، فلا يمكن تغطية جميع الحالات بسبب تعقيد بعض تطبيقات الويب، وبالتالي هذه الاختبارات غير قادرة على كشف العيوب المستقبلية التي قد تحصل خصوصا مع توسع التطبيق وزيادة عدد المستخدمين، لهذا يكون الاختبار اليدوي أقل دقة وموثوقية مقارنةً بالاختبار الآلي.
2- Automation testing (الاختبار الألي): في هذه المنهجية يتم الاختبار بواسطة تقنيات وتطبيقات متنوعة، بدلاً من إجراء الاختبارات اليدوية التي قد تسبب بعض النتائج غير المرغوب فيها والتي تكون مكلفة وبطيئة، تميل الشركات إلى استخدام أدوات الاختبار مفتوحة المصدر أو التجارية التي تجعل طريقة الاختبار أسهل وأفضل بالنسبة لهم، وتوفر التكلفة، وتكسبهم المزيد من الوقت، وتحقق دقة أعلى في قياس معايير الأداء (زمن الاستجابة، زمن العرض، الإنتاجية، قابلية التوسع...) في ظروف حمل معينة، وبالتالي اكتشاف العيوب قبل أن تحدث، ولكن من مساوئ هذه المنهجية، أنه لا يوجد أداة قادرة على إجراء كافة الاختبارات المتعلقة بقياس معايير الأداء، وبالتالي لا يمكن الاعتماد على إحداها بشكل منفصل، حيث ذكرت الأبحاث أن لكل أداة مميزات تجعلها قادرة على إجراء الاختبارات وكشف عيوب معينة في الأداء، بالتالي نحتاج الى بناء تكامل فيما بينها.

2.1. أهمية البحث:

يهدف البحث إلى بناء إطار عمل يحقق التكامل بين الأدوات والتقنيات المتاحة للاختبار، ومساعدة شركات البرمجة في اختبار تطبيقاتها بشكل كامل، واكتشاف عيوب الأداء قبل أن تحدث وخاصة ضمن سويات مختلفة من الحمل، وتجنب الفحص اليدوي أو الاعتماد على أداة دون أخرى، وتقديم نتائج الاختبارات إلى مطوري التطبيقات بهدف الاستفادة منها لزيادة رضا المستخدمين، وتحقيق إنتاجية أعلى للشركات.

3.1. هيكلة الدراسة:

تم تقسيم هذه الدراسة إلى مبحثين، يتناول المبحث الأول منها الإطار النظري للدراسة والدراسات السابقة، بينما يتطرق المبحث الثاني إلى الإطار العملي وتطبيق عملي يعتمد على الإطار المقترح.

2.المبحث الأول- الإطار النظري والدراسات السابقة

1.2. الإطار النظري اختبارات الأداء مفاهيم وقضايا

1.1.2. الهدف من اختبار الأداء: تهدف الشركات إلى توفير المعلومات المتعلقة بالسرعة والقابلية للتوسع واستقرار التطبيق قبل الإصدار، مما يتيح اتخاذ قرارات صحيحة تساعد على انطلاق التطبيق على مستوى عالي من الأداء، يساعد اختبار الأداء فيما يلي

1- الكشف عن مشاكل الأداء في وقت مبكر من دورة حياة المشروع، وبالتالي تقليل التكلفة والجهد.
2- لتحسين قدرة التطبيق على استيعاب الحمل المتوقع، من خلال العمل على اكتشاف الأماكن التي يجب تحسينها، مثلا اكتشاف أن التطبيق يحتاج إلى زمن كبير من أجل تحميل صفحة معينة.

3- للتحقق من الموثوقية (إمكانية استقرار التطبيق في كافة الحالات المستقبلية المتوقعة).
4- تحقيق الوصول الى اتفاقيات مستوى الخدمة (SLAs) المحددة مسبقًا للأداء.
5- فهم استخدام الموارد والإنتاجية ويساعد في تحديد الاستخدام الأمثل للموارد من أجل تحقيق أعلى إنتاجية.
6- تقييم الأداء المتصور من وجهة نظر المستخدم النهائي عبر مختلف المتصفحات.
7- تحديد عبء العمل (عدد الطلبات التي يعالجها النظام في الثانية).
8- تحديد أوقات الاستجابة المختلفة في جميع الطبقات.
9- معرفة استهلاك التطبيق للموارد على المخدمات.
10- تحليل عنق الزجاجة لتحديد أي اختناقات يمكن أن تحصل في المستقبل. (Shivakumar, 2020) (Kyrychenko, 2021)

2.1.2. أنواع اختبارات الأداء: ورد ضمن المراجع (Srivastava, 2021) (Shivakumar, 2020) (Matam, 2017)

1. اختبار الحمل (Load Tests): يتم إجراء اختبارات تحميل بمستويات حمل مختلفة لمعرفة سلوك التطبيق، يتم تسجيل التغيير في سلوك أداء التطبيق مع الزيادة في الحمل أثناء الاختبار.
2. اختبار الاجهاد (Stress Tests): يتم إخضاع التطبيق إلى حمل زائد وبعد ذلك تتم ملاحظة ثباته وأدائه، وقياس الأداء والسلوك وبالتالي يمكن تحديد نقطة توقف التطبيق وتحديد أقصى مستويات الحمل التي يمكن للتطبيق التعامل معها دون انهيار الأداء.
3. اختبار التحمل (Endurance testing): يتم إخضاع التطبيق لحمل محدد لمدة طويلة (عادة 3-72 ساعة) لقياس أداء وسلوك التطبيق، بحيث يمكن تحديد المشكلات المتعلقة بالأجهزة، منها تسريب محتمل للذاكرة.
4. اختبار أقصى حمل (Peak Load Tests): على سبيل المثال، تشهد مواقع التجارة الإلكترونية على شبكة الإنترنت ذروة حركة المرور خلال أيام محددة مثل أيام عيد الميلاد، ولذلك فإن هذا الاختبار سيختبر التطبيق ضمن ظروف حمل عالية متوقعة في بعض الأوقات.
5. اختبار القابلية للتوسّع (Scalability Tests): يتم تعريف قابلية التوسع على أنها مدى تعامل التطبيق مع الزيادة في الحمل مع الاستمرار في تلبية معايير الأداء المطلوبة، يتم هذا الاختبار باستخدام أحمال عمل مختلفة في نموذج عبء العمل، حيث يتم زيادة الحمل بشكل تدريجي تبدأ من 10٪ من البيانات، وتصل إلى الحد الأقصى باستخدام زيادة حجم البيانات بشكل دوري ومراقبة أداء التطبيق.
6. اختبار تغير الحمل المفاجئ (Spike testing): يتم في هذا الاختبار تطبيق زيادة مفاجئة في الحمل ومراقبة استجابة وسلوك التطبيق.
7. اختبارات السعة (Capacity Tests): هو اختبار يحدد أقصى حمل يمكن للتطبيق التعامل معه مع مراعاة تحقيق معايير الأداء المطلوبة، ويسمى القياس الناتج السعة القصوى، وهي تستخدم في توسيع نطاق التطبيق وتقدير تكاليف النمو في المستقبل.
8. اختبارات التوفر العالي/المخدم الاحتياطي (High Availability Test/Fail-Over Tests): تم تصميم البنية التحتية الحديثة لتطبيقات الويب لتكون متاحة بدرجة كبيرة ومرنة على مواجهة أعطال الأجهزة والبرامج، من الناحية المثالية، يجب أن تضمن البنية عدم وجود نقطة فشل واحدة وأن هناك مخدمات احتياطية يمكنها أن تؤدي المطلوب دون التأثير على تجربة المستخدم، في هذا الاختبار، تتم محاكاة العديد من أعطال

المعدات والبرامج، ويتم تشغيل اختبارات الأداء ذات الصلة للتحقق من أن التطبيق لا يزال يستوفي معايير الأداء.

3.1.2. مراحل اختبار الأداء: ورد ضمن المرجع (Shivakumar, 2020)عدة نشاطات لإنجاز اختبار الأداء وهي

1. تقييم الأداء وتحديد المتطلبات: يتم جمع متطلبات اختبار الأداء باستخدام وثيقة المتطلبات النهائية من خلال:
   - أ- التعرف على الوظائف الأساسية للتطبيق.
   - ب- تحديد احتياجات اختبار الأداء.
   - ج- التحقق من المتطلبات.
   - د- تحديد مميزات المستخدم.
   - هـ- تحديد مقاييس الأداء، مثل متوسط زمن الاستجابة.
   - و- التعرف على المتطلبات غير الوظيفية (NFR) واتفاقيات مستوى الخدمة.
   - ز- تحديد تحديات الأداء الحالية.
2. تخطيط اختبار الأداء: يقوم فريق الاختبار بتطوير استراتيجية لاختبار الأداء وفق الخطوات التالية
   - أ- إعداد بيئة اختبار الأداء.
   - ب- تطوير استراتيجية اختبار الأداء وتحديد الحمل المراد تطبيقه على التطبيق.
   - ج- تحديد أدوات اختبار الأداء.
3. برمجة اختبار الأداء:

   خلال هذه المرحلة، يتم برمجة حالات الاختبار بناء على ما تم دراسته بالمرحلة السابقة.
4. تنفيذ اختبار الأداء: يوضح الجدول (1) دورات تنفيذ اختبار الأداء

الجدول (1) دورات تنفيذ اختبار الأداء

دورة التنفيذ	الوصف
دورة مستخدم واحد	يتم تنفيذ هذا الاختبار من دون حمل بهدف اختبار تجربة مستخدم واحد فقط، ويتم تسجيل جميع المقاييس المطلوبة.
دورة تحميل 20%	يتم تنفيذ هذا الاختبار باستخدام 20٪ من ظروف الحمل القصوى، ويتم بالمراحل الأولى من دورات التنفيذ، ويتم تسجيل جميع المقاييس المطلوبة.
دورة أقصى حمل	يتحقق من صحة أداء التطبيق في ظل ظروف تحميل الذروة من ساعة إلى ساعتين، بهدف معرفة أوقات استجابة واستخدام التطبيق للموارد.
دورة عتبة الاجهاد	يتم زيادة نشاط المستخدم بشكل تدريجي إلى حمل يعادل 150٪ -200٪ من حمل المستخدم الاقصى، حتى الوصول إلى زمن استجابة غير مقبول، يتم تسجيل أوقات الاستجابة ومعرفة عتبة انهيار التطبيق.
دورة الثبات (التحمل)	يتم تطبيق حمل ثابت أقل من السعة القصوى للتطبيق لمدة12 إلى 24 ساعة وذلك بهدف اكتشاف مشاكل تسرب الذاكرة.

5. تقرير النتائج والتوصية: يتم كتابة تقرير مفصل عن نتائج حالات الاختبار يتضمن
   - أ- توثيق نتائج الاختبار والمقاييس التي تم اعتمادها.
   - ب- توصيات باتخاذ بعض القرارات المتعلقة بتحسين الأداء.

4.1.2. بنية أدوات الاختبار: ورد ضمن المرجع (Molyneaux, 2014)مكونات أدوات الاختبار الالية وهي

1- وحدة تسجيل نشاط المستخدم (Scripting module): يمكن من خلال هذه الوحدة تسجيل نشاط المستخدم النهائي، وتدعم العديد من البروتوكولات المختلفة وذلك حسب الأداة، ويمكن التحكم بالنشاطات المسجلة وتعديلها.

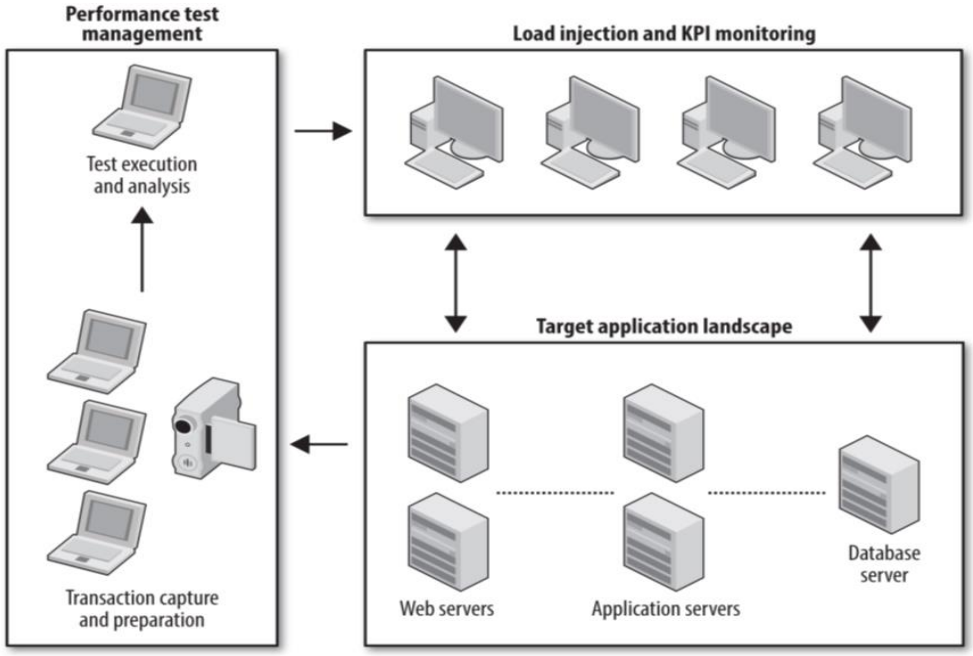
2- وحدة إدارة الاختبار (Test management module): تمكن هذه الوحدة من إنشاء وتنفيذ حالات اختبار الأداء.

3- مولد الحمل (Load injector): يتم من خلالها إنشاء عدد من المستخدمين الافتراضين بحسب ما هو مطلوب لإنجاز حالة الاختبار.

4- وحدة التحليل (Analysis module): يمكن من خلال هذه الوحدة تحليل النتائج التي تم استخلاصها من كل حالة اختبار، يمكن أن تكون البيانات على شكل تقرير أو رسم بياني أو عرض جدولي.

5- وحدات اختيارية (Optional modules): يمكن أن تحتوي الأداة على وحدات تسمح بالتكامل مع أداة أخرى، أو لإضافة ميزات جديدة للأداة، يوضح الشكل رقم (4.2) بنية الأداة النموذجية لاختبار الأداء.

يوضح الشكل (1) بنية أداة الاختبار النموذجية



الشكل (1) بنية أداة الاختبار النموذجية

2.2. الدراسات السابقة:

1- **ذكر الباحثون من خلال البحث (Jha, 2017)** أهمية اختبار الأداء من أجل ضمان خصائص الجودة لتطبيق الويب، حيث أن التحدي الأساسي في اختبار الأداء ليس اكتشاف الأخطاء بل التخلص من اختناقات الأداء، وتعتبر أن JMeter الأداة المفتوحة المصدر هي الوسيلة لاختبار الحمل وقياس الأداء، حيث تم دراسة أداء موقعين للتجارة الالكترونية، وقدمت تقرير عن نتائج اختبارات أداء مواقع التسوق على الويب، والتي تتضمن مخططات رسومية لمتوسط معدل الاستجابة والإنتاجية والانحراف المعياري، وأثبتت هذه النتائج تفوق إحدى الموقعين على الاخر بسبب زيادة الإنتاجية وانخفاض سرعة الاستجابة، والذي سيؤدي الى مزيد من رضا العملاء عن غيره من المواقع الإلكترونية.

2- **أكد الباحثون من خلال البحث (Pradeep, 2019)** على أن عملية اختبار البرمجيات إلزاميه سواء كان تطبيق ويب أو تطبيق سطح مكتب، وذلك قبل إطلاق التطبيق للخدمة بهدف التنبؤ بسلوكه، وضمان عدم وجود عيوب، وذكر الباحثون ضمن هذا البحث العديد من الأدوات المستخدمة في اختبارات الأداء، وتم اختيار ثلاث أدوات هي Locust, Apache JMeter, Hulk Automation لإجراء اختبار الاجهاد ومقارنة النتائج، ومن أجل تقييم أداء أدوات الاختبار تم اختبار مواقع الويب المختلفة بنفس الحمل وحركة المرور بحيث يمكن حساب زمن التنفيذ والإنتاجية، وأظهرت النتائج تفوق Locust ثم يأتي بعدها، Hulk Automation ثم Apache JMeter من حيث زمن تنفيذ الاختبار. اتبع الباحثون المنهجية التالية في عملية الاختبار

- توليد حمل عشوائي.
- تطبيق هذا الحمل على السيرفر.
- تحليل عوامل الأداء.
- تسجيل النتائج.
- تقييم الأداء مع الحمل (حركة المرور المولدة).

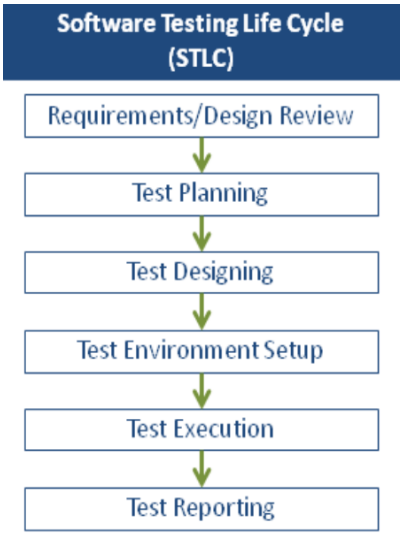
3- **ورد ضمن البحث (Sasankar, 2019)** المراحل الأساسية من أجل أداء الاختبارات والمشار اليها في الشكل رقم (2)، وذكر الباحثون أن أدوات إدارة الاختبار تساعد في جمع بيانات التنفيذ من الاختبارات الآلية، وإدارة بيئات متعددة وإدخال معلومات حول العيوب الموجودة، كما أنها توفر إمكانية تبسيط عملية الاختبار وتتيح الوصول السريع إلى تحليل البيانات وسهولة الاتصال عبر فرق المشروع المتعددة.

أ- من بين الأدوات المستخدمة في إدارة علميات الاختبار:

- (Test Environment Toolkit) TET: كان الهدف من إنشاء مجموعة أدوات بيئة الاختبار (TET) هو إنتاج أداة لقيادة التحكم في اختبارات الأداء.
- TETware: يوفر إطار اختبار موحد متعدد المنصات سهل الاستخدام يمكن دمج مجموعات الاختبار المحلية والبعيدة والموزعة وفي الوقت الحقيقي فيها.
- Test Manager: يُستخدم لتسهيل أنشطة تطوير البرامج العادية، وأتمتة أنشطة الاختبار وإدارتها.
- (Requirements and Testing Hub) RTH: يوفر تسهيلات تتبع الأخطاء.

ب- أدوات اختبار الحمل: يتم إجراء اختبار الحمل لتحديد سلوك النظام في ظروف حمل الذروة المتوقعة.

- JMeter: تطبيق Java مصمم لإجراء الاختبارات المتعلقة بقياس معايير الأداء.
- FunkLoad: مصمم بلغة بايثون، من أجل اختبارات الأداء والإجهاد.

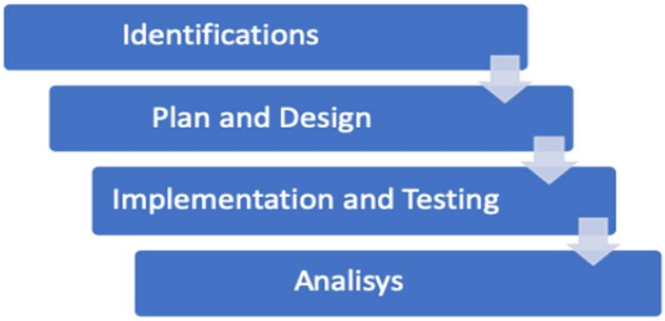


الشكل رقم (2) مراحل اختبار البرمجيات

4- **قدم البحث (Wang, 2019)** دراسة حول اختبار أداء نظام إدارة الجامعة، يشمل عدد كبير من المستخدمين (الموظفين الإداريين، وموظفي الأقسام والطلاب)، وتهدف هذه الدراسة إلى مساعدة المطورين في تحليل ما إذا كان أداء النظام أثناء تسجيل الدخول والخروج يلبي الاحتياجات الحالية والمستقبلية، من أجل ضمان تجربة مستخدم جيدة، يجب أن يكون متوسط زمن الاستجابة أثناء تسجيل الدخول والخروج هو (4) ثانية، وعند تطبيق حمل 500 مستخدم متزامن، أصبح متوسط زمن الاستجابة (7.483)، حيث أكد الباحثون أن هذه النتائج توفر مرجعا لمطوري ومحللي النظام من أجل معالجة اختناقات أداء النظام، وتوفير الحلول المناسبة لها.

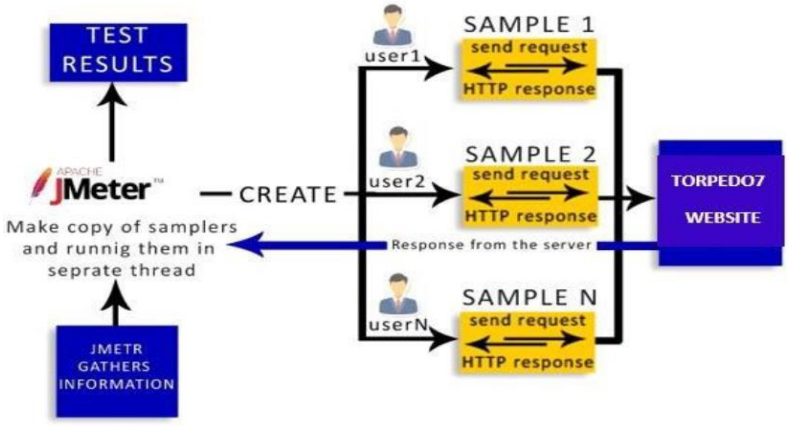
5- **قدم الباحثون في البحث (Maila-Maila, 2019)** دراسة عن البرمجيات المفتوحة المصدر، وحللوا قدرات كل برنامج فيما يتعلق بتدابير الجودة في الأداء التي تنظمها المعايير الدولية واعتبروا أنه من الضروري تحديد عملية لتنفيذ اختبارات تحدد جودة منتج تطبيق الويب فيما يحقق رضا وقبول المستخدم، وقد أجريت هذه الدراسة على أساس مقاييس المعيار ISO/IEC 25023، وعملية اختبار البرمجيات المقترحة من قبل ISTQB والتي تحدد معايير الخصائص الفرعية وتدابير الجودة لخصائص الأداء، أكدت الدراسة أن الأدوات مفتوحة المصدر لتقييم أداء تطبيقات الويب هي بديل جيد للبرامج الخاصة، وبعد التحقق من الأدوات المتاحة وتجريبها كانت الأداة التي حصلت على أفضل النتائج هي JMeter، الذي يغطي 80% من معايير الاختبار.

6- **قدم الباحثون من خلال البحث (Hidayanto, 2019)** المنهجية المشار إليها بالشكل رقم (3) لاختبار أحد مواقع الدفع الالكتروني في اندونيسيا باعتبار أن عدد المستخدمين كبير لهذا الموقع، الغرض من هذه الدراسة هو تحليل أداء مخدمات Nginx وApache باستخدام JMeter عند زيادة عدد المستخدمين، وقد أظهرت النتائج أن زمن الاستجابة في مخدمات Nginx أقل من مخدمات Apache أما بالنسبة لزمن التأخير فإن مخدمات Apache تتفوق على Nginx والإنتاجية في مخدمات Nginx أكبر من مخدمات Apache في حال عدد مستخدمين كبير، أما عدد مستخدمين قليل تكون إنتاجية Apache أفضل، والسبب وراء أداء Nginx هو، البنية المصممة لـ Nginx، وهي بنية قائمة على الأحداث حيث يتم معالجة كل شيء فقط في عملية رئيسية واحدة تستخدم الذاكرة المشتركة، في Apache Server، سيتم إنشاء عملية لكل اتصال تستخدم الذاكرة الخاصة بها وبالتالي ميزة استخدام Nginx أي خادم غير متزامن هو قابلية التوسع، تقدم Apache كفاءة أفضل من خلال خدمة حمل محدود في حين أن Nginx أقل كفاءة.



الشكل رقم (3) منهجية لاختبار الأداء

7- **ورد ضمن البحث (Suryadevara, 2020)** أن النمو الاقتصادي لأي منظمة أو شركة يعتمد على تطبيق الويب الخاص بها، في حالة أن التطبيق لا يلبي معايير الأداء وخاصة المتعلقة بسرعة استجابة التطبيق للمستخدم، يفقد المستخدم اهتمامه بهذا التطبيق، وبالتالي مزيد من الخسائر للشركات والمنظمات، لذلك لابد من إجراء اختبار الأداء من أجل التنبؤ بسلوك المستخدم في حال وجود حمل مطبق على التطبيق، والحصول على معلومات دقيقة عن تجربة المستخدم وموثوقية التطبيق، اختار الباحثون الموقع torpedo7.co.nz من أجل محاكاة تجربة المستخدم في أيام عيد الميلاد ويوم الجمعة، وتم التأكيد أن بيئة الاختبار يجب أن تكون مماثلة لبيئة الإنتاج، لذلك اقترحوا إجراء الاختبار بعد الساعة 10 مساء حيث يكون عدد المستخدمين الفعليين قليل، ويكون الحمل المطبق على الموقع هو نفس الحمل المطبق من أداة الاختبار وبالتالي نحصل على نتائج دقيقة، وكانت نتائج الاختبار أن الموقع مثالي لمعالجة الزيادة في الحمل لحدود 500 مستخدم، ولا يحتاج إلى ترقيات، قد استخدم الباحثون المنهجية المشار إليها في الشكل رقم (4) لاختبار صفحات تطبيق الويب.

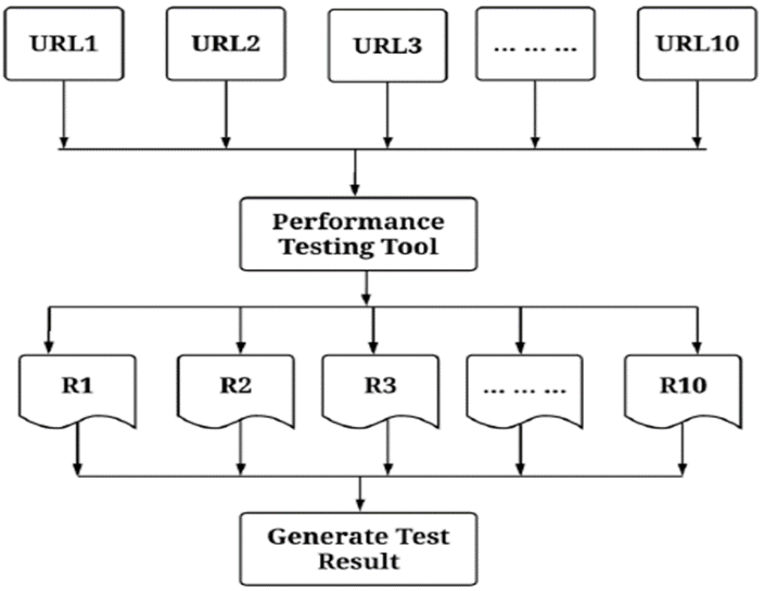


الشكل رقم (4) منهجية اختبار باستخدام JMeter

8- **قدم البحث (Hossain, 2021)** إطارا لاختبار الأداء معتمدا على ثلاث أدوات تقيس (زمن التحميل، والزمن حتى البايت الأول، وزمن بداية العرض، زمن أول عرض للمحتوى، زمن عرض أكبر محتوى، زمن الحظر، مؤشر السرعة، التغيير في تخطيط عرض الصفحة، والزمن اللازم للتفاعل) ثم يتم حساب المتوسط الحسابي لنتائج الأدوات الثلاثة.

اختار الباحثون أفضل 10 مواقع للتجارة الالكترونية في بنغلادش، ولم يتم ذكرها بشكل صريح من أجل الحفاظ على خصوصيتها، ومن خلال نتائج الاختبارات التي أجريت ذكر الباحثون أن الموقع السابع أقل زمن حظر بين المواقع العشرة، وكذلك سجل الموقع العاشر أعلى زمن تحميل، مما يشير إلى أن هذا الموقع يستغرق وقتًا أطول في التحميل مقارنة بالآخرين.

يتميز هذا الإطار بقدرته على كشف العيوب الموجودة في صفحة الويب في حال عدم وجود أية سوية حمل يخضع لها التطبيق، ولا يمكن التنبؤ بزمن الاستجابة في سويات مختلفة من الحمل لأن الأدوات الداخلة في تكوين الإطار وهي Webpage Test, PageSpeed Insights, Gtmetrix لا تدعم تغيير سويات الحمل، يوضح الشكل (5) بنية الإطار المقترح.

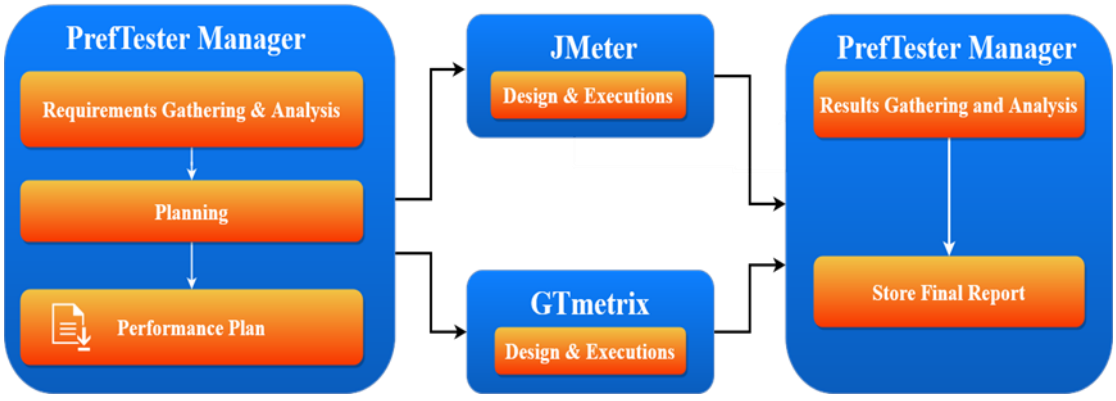


الشكل (5) بنية الإطار.

3.المبحث الثاني- الإطار العملي

1.3. الإطار المقترح:

بعد مراجعة مفاهيم اختبارات الأداء الواردة ضمن المبحث الأول تم اقتراح إطار PrefTester، يوضح الشكل رقم (6) مخطط صندوقي للإطار المقترح.



الشكل (6) مخطط صندوقي للإطار المقترح.

يتكون الإطار المقترح من الأدوات التالية

- JMeter: أداة مجانية مفتوحة المصدر يمكن من خلالها تنفيذ سيناريوهات متعددة تحتوي سويات مختلفة من الحمل، وبالتالي التنبؤ بسلوك التطبيق خلال هذه السويات.
- GTmetrix: أداة مجانية يمكن من خلالها كشف المشاكل المتعلقة بأداء وبنية صفحة الويب.

- PrefTester Manager: أداة تم تطويرها باستخدام إطار العمل Laravel وتم من خلالها تحقيق المنهجيات المطلوبة في مرحلة جمع المتطلبات والتخطيط وجمع وتقرير النتائج والتوصيات.

وسيتم تحقيق كافة مراحل الاختبار بواسطة هذه الأدوات الثلاثة كما يلي:

1.1.3. مرحلة جمع المتطلبات: يتم في هذه المرحلة جمع متطلبات الاختبار، وفهم وظائف التطبيق الأساسية، وبسبب تعقيد الويب، ومن أجل بناء خطة اختبار تحاكي تجربة المستخدم الحقيقي، تم تحديد المتطلبات في أداة إدارة عملية الاختبار وفق ما يلي:

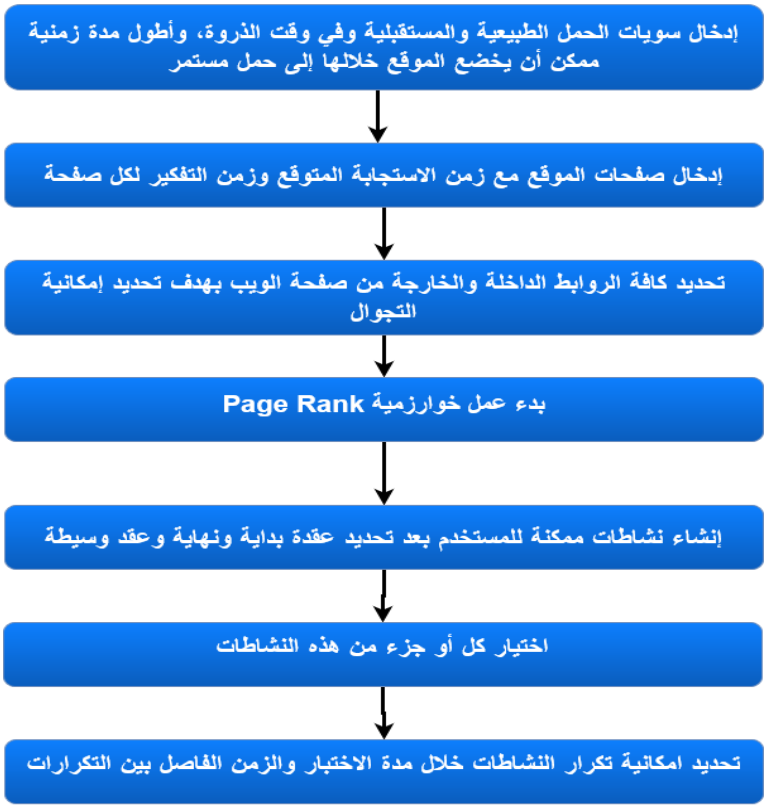
1- إدخال عدد المستخدمين المتوقع خلال مدة زمنية محددة من أوقات الحمل الطبيعية.
2- إدخال عدد المستخدمين خلال مدة زمنية محددة من أوقات حمل الذروة المتوقعة.
3- إدخال عدد المستخدمين خلال مدة زمنية من أوقات الحمل الطبيعية متوقع أن يصل لها التطبيق خلال المستقبل.
4- إدخال أطول مدة زمنية يمكن أن يخضع التطبيق خلالها إلى حمل مستمر.
5- تسجيل زمن الاستجابة المتوقع لكل صفحة ويب.
6- تسجيل زمن التفكير لكل صفحة ويب، وهو الزمن الذي يستغرقه المستخدم الحقيقي ضمن الصفحة قبل الانتقال للصفحة التالية.
7- معرفة إمكانية تجوال المستخدم ضمن موقع الويب من خلال:
   أ- تسجيل كافة الصفحات التي تؤشر إلى صفحة الويب.
   ب- تسجيل كافة الصفحات التي تشير عليها صفحة الويب.
8- تحديد أهمية كل صفحة من صفحات الموقع باستخدام خوارزمية (PageRank) وفق العلاقة التالية:

$$PR(u) = c \sum_{v \in B(u)} \frac{PR(v)}{N_v}$$

- U: تمثل صفحة ويب.
- Bu: تمثل مجموعة الصفحات التي تشير للصفحة U.
- PR(u): يمثل ترتيب الأهمية (rank) للصفحة U.
- PR(v): يمثل ترتيب الأهمية (rank) للصفحة V.
- Nv: يمثل عدد الروابط الخارجة من الصحفة V.
- C: يستخدم للإشارة إلى normalization. (Joshi, 2020)

9- مساعدة المختبر في استنتاج كافة النشاطات (Iterations) -النشاط يمثل مجموعة من الصفحات المتوقع زيارتها بشكل تسلسلي- المتوقعة من قبل المستخدم الحقيقي ضمن موقع الويب، حيث يقوم المختبر بتحديد الصفحات المراد تكوين النشاطات من خلالها مع تحديد صفحة البداية والنهاية، وبعدها يقوم المختبر باختيار النشاطات المرشحة لتوليد نموذج الحمل المتوقع وفق ما يناسب حالات الاختبار.
10- تحديد عدد مرات تكرار المستخدم للنشاط خلال مدة الاختبار، وتحديد Pacing time وهو مقدار من الوقت يتم استخدامه في اختبارات الأداء للتحكم في معدل إرسال وتنفيذ الطلبات على المخدم (Molyneaux, 2014).

يوضح الشكل رقم (7) خطوات تحضير متطلبات الاختبار.



الشكل رقم (7) خطوات تحضير متطلبات الاختبار

2.1.3. مرحلة تخطيط اختبار الأداء: تعتبر عملية التخطيط من المراحل الهامة لاختبار الأداء، حيث يجب بناء خطة تحاكي تجربة المستخدم في كافة أنواع الاختبارات بناء على المتطلبات التي جمعها في المرحلة السابقة، حيث يتم من خلال PrefTester Manager في هذه المرحلة بناء نموذج الحمل المتوقع وفق الخطوات التالية:

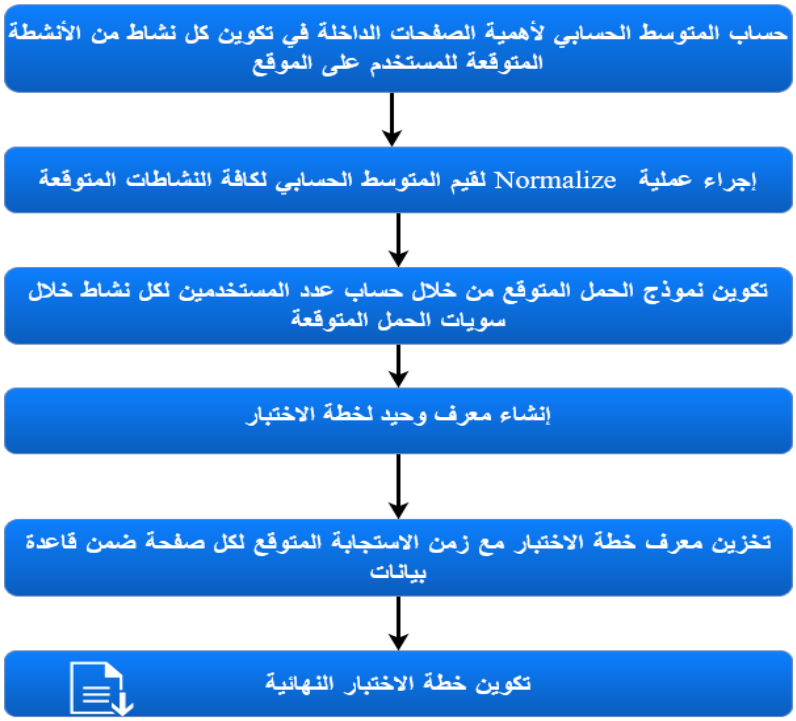
1- يقوم الإطار بحساب أهمية النشاط المرشح (Iteration) وذلك من خلال إيجاد المتوسط الحسابي لأهمية الصفحات المكونة له.
2- يتم إجراء عملية Normalize للمتوسط الحسابي لأهمية النشاطات المرشحة، وبناء علية يتم توزيع الحمل المتوقع في كافة السيناريوهات على النشاطات المولدة حسب أهميتها.

يقدم الإطار المقترح خطة الاختبار على شكل مستند PDF أو صفحة ويب ضمن أداة إدارة الاختبار، تتضمن هذه الخطة تفاصيل شاملة عن عملية الاختبار، تمكن المختبر من فهم المتطلبات والأدوات اللازمة لعملية الاختبار وهي:

1- هدف ومجال خطة الاختبار.
2- أدوات وبيئة الاختبار.
3- سيناريوهات الاختبار.
4- محددات كل سيناريو من سيناريوهات الاختبار وتتضمن:
أ- الهدف من السيناريو.
ب- والأداة المستخدمة.
ج- المنهجية.
د- النتائج المتوقعة.

5- نموذج الحمل المتوقع.

نلاحظ أن الإطار في هذه المرحلة ومن خلال PrefTester Manager يقوم ببناء خطة الاختبار بعد أن تم جمع المعلومات المطوبة عن الموقع (صفحات الموقع كاملة، سويات الحمل الطبيعية، وفي وقت الذروة، وسويات الحمل المستقبلية خلال مدة زمنية، ويتم حساب سوية الحمل الموافقة لأطول مدة زمنية من خلال إجراء تناسب مع سوية الحمل المتوقعة في الأوقات الطبيعية، الروابط الداخلة والخارجة من كل صفحة من أجل تحديد أهمية الصفحة من خلال خوارزمية Page Rank، لأن الصفحات لا تخضع لنفس الحمل حيث يوجد صفحات تكون بوابة لصفحات أخرى وتشكل عنق الزجاجة ضمن الموقع)، ثم يتم بناء نموذج الحمل المتوقع ومن خلاله يتم توزيع عدد المستخدمين ضمن سويات الحمل على الصفحات حسب أهميتها، وبالتالي المساعدة في اكتشاف الصفحات التي قد تتعرض لعدد مستخدمين أكثر من غيرها وقد تسبب عنق الزجاجة، وهذا ما يميز الإطار عن المنهجيات السابقة التي كانت تعتمد على تطبيق الحمل بشكل متساوي لجميع الصفحات، ودون الاعتماد على تحليل النشاطات الممكنة للمستخدم على الموقع، أما خلال الإطار نجد أن تحليل الموقع قبل الاختبار يتم من خلال أداة الإدارة، وبالتالي تقتصر مهمة المختبر فقط بتزويد الأداة بالبيانات التي تم ذكرها أعلاه لإنتاج خطة شاملة لجميع الحالات التي قد يخضع قد المستخدم خلال الواقع.



الشكل رقم (8) خطوات بناء خطة الاختبار

3.1.3. مرحلة برمجة اختبار الأداء: في هذه المرحلة يتم تهيئة السيناريوهات المقترحة ضمن الخطة للتنفيذ، وذلك باستخدام نمط الواجهة الرسومية (GUI mode) ل JMeter وفق الخطوات التالية:

1- تهيئة إعدادات (Proxy) على المتصفح وJMeter، بهدف تسجيل كافة الصفحات.
2- مسح ذاكرة التخزين المؤقت ضمن المتصفح.
3- إضافة (Thread Group) لكل Iteration ضمن نموذج الحمل المقترح.
4- إضافة (Transaction Controller) لكل صفحة ويب ضمن Iteration.

5- تسجيل النماذج السابقة باستخدام (HTTP(S) Test Script Recorder) ضمن JMeter من خلال تسجيل كل صفحة في نموذج الحمل داخل (Transaction Controller).

6- التحقق من نجاح عملية تسجيل الخطوات السابقة.

7- تهيئة (Thread Group) حسب الحاجة من العناصر التالية:

- Timers: تستخدم بهدف تحقيق وقت التفكير لكل صفحة ويب.
- Flow Control Action sampler: الغرض من استخدامه لتحقيق Pacing المحددة ضمن نموذج الحمل المقترح.
- CSV Data Set Config: بهدف تمرير متغيرات إلى صفحة الويب المراد اختبارها.

8- يتم حفظ السيناريو كملف JMX للتنفيذ في المرحلة اللاحقة.

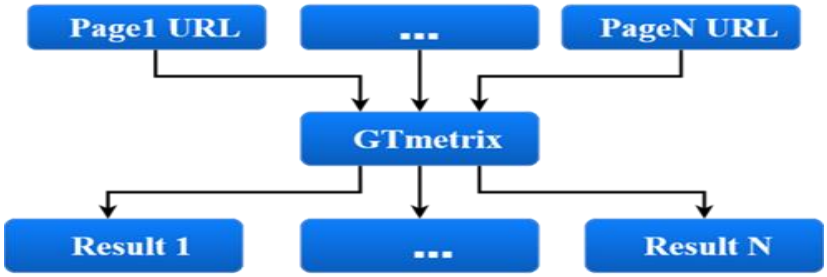
4.1.3. مرحلة تنفيذ اختبار الأداء: يتم ضمن الإطار المقترح تنفيذ خطة الاختبار بمساعدة الأدوات المقترحة، حيث يتم تنفيذ السيناريو الأول الخاص ب GTmetrix كما هو وارد في خطة الاختبار وفق ما يلي:

1- اختيار متصفح ويب.

2- الدخول إلى عنوان https://gtmetrix.com/.

3- إدخال عنوان الصفحة المراد اختبارها.

يوضح الشكل (9) منهجية الاختبار بواسطة GTmetrix.



الشكل رقم (9) منهجية الاختبار بواسطة GTmetrix.

تدمج GTmetrix بين مدى سرعة أداء صفة الويب (من حيث التحميل والتفاعل والاستقرار البصري) ومدى جودة تصميمها (بنيتها) لتحقيق الأداء الأمثل، حيث يتم تخصيص 70% من التقييم لمعايير الأداء و30% لمعايير البنية، وفق التقييم التالي:

أ- تقييم أداء التحميل من خلال

- مؤشر أول عرض للمحتوى.
- سرعة الصفحة.
- مؤشر أكبر محتوى.

ب- تقييم التفاعلية من خلال

- الزمن اللازم للتفاعل.
- زمن حظر العرض.

ج- تقييم الاستقرار البصري من خلال

- الازاحة في تخطيط صفحة الويب.

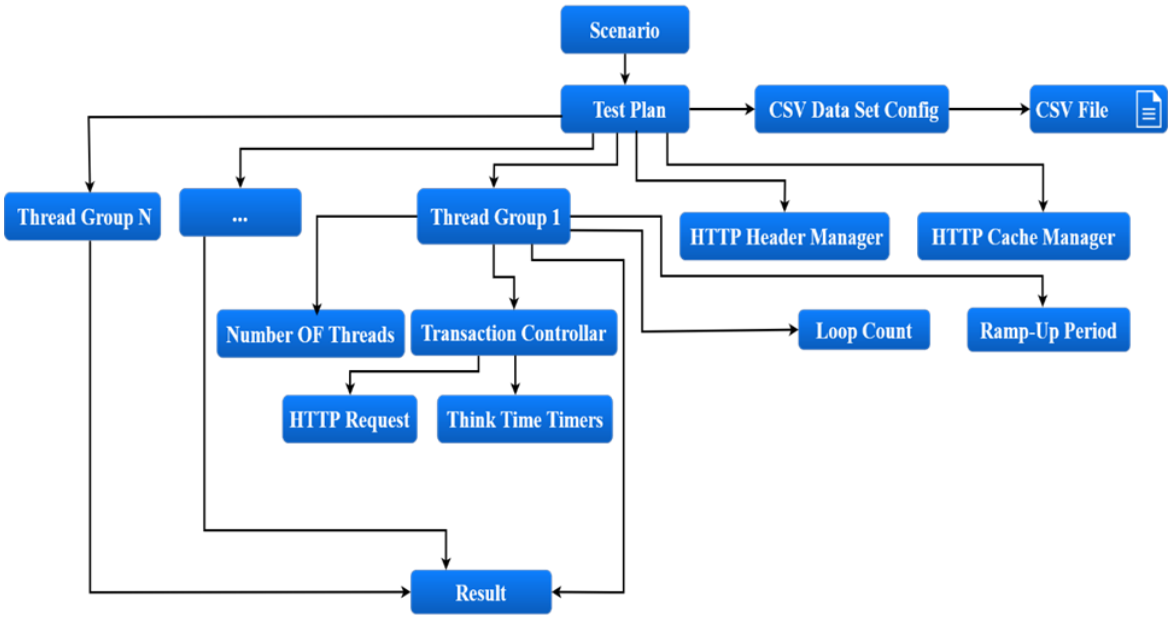
د- تقييم بنية صفحة الويب من خلال

- تمكين Keep-Alive ويتم التحقق من هذا المقياس للصفحات التي تستخدمHTTP/1.1.
- التحقق من جمع الصور باستخدام CSS sprites.
- التحقق من استخدم شبكة توصيل المحتوى (CDN).
- التحقق من تضمين ملفات ال css وهذا يسبب تحميل الملفات بشكل تسلسلي بدلاً من تحميلها على التوازي، وبالتالي التأخر في عرض المحتوى على صفحة الويب. (https://gtmetrix.com/)

أما السيناريوهات المحددة للتنفيذ بواسطة JMeter وهي (الثاني والثالث والرابع والخامس والسادس) تنفذ من خلال موجة الأوامر (CLI Mode) وذلك من أجل الحصول على نتائج مثالية للاختبار وأكثر دقة، وذلك من خلال الأوامر التالية:

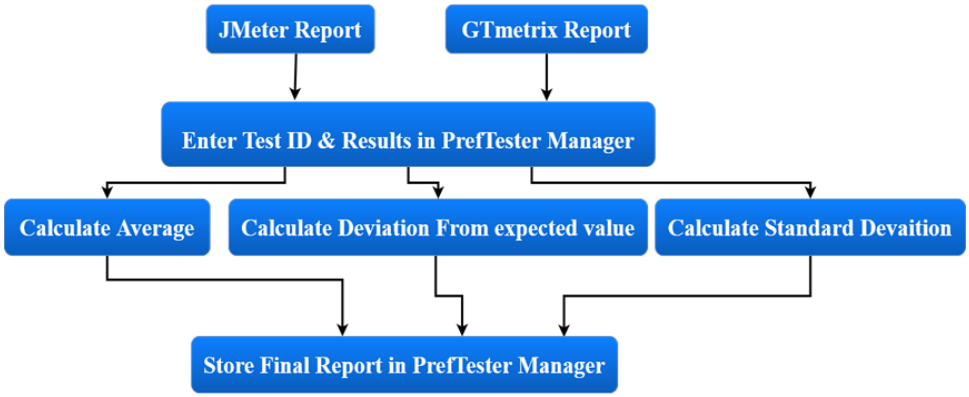
- -n لتحديد نمط عمل JMeter (NON GUI mode).
- -t لتحديد مسار واسم ملف JMX الذي يحتوي على سيناريو الاختبار.
- -l لتحديد مسار واسم ملف نتائج الاختبار.
- -e لتوليد ملف HTML لنتائج الاختبار.
- -o لتحديد مسار ملف الHTML المولد عن الاختبار. (https://jmeter.apache.org)

يوضح الشكل رقم (10) منهجية الاختبار بواسطة JMeter، وبمقارنة هذه المنهجية مع المنهجيات الواردة ضمن الدراسات السابقة، نجد أنها تحتوي على عناصر تحكم إضافية تجعل بيئة الاختبار أقرب ما يمكن للواقع، حيث أن كل Thread Groupيماثلها نشاط في نموذج الحمل المقترح، وكل Transaction Controller يماثل صفحة ضمن النشاط، ولكل صفحة عنوان URL يمثله HTTP Request وزمن التفكير يمثله Think Time Timers، وعدد مرات تكرار النشاط يمثله Loop Count، وقيمة Ramp-up Period تكون ثابتة لكل النشاطات ضمن السيناريو ويتم ذكر قيمتها ضمن محددات سيناريو الاختبار، وقد تم جمعها في مرحلة جمع المتطلبات لكل سوية من سويات الحمل المطلوبة للاختبار. وتم استخدام HTTP Header Manager حتى يتمكن JMeter من تخصيص المعلومات التي يرسلها ضمن Header Request لكل طلب HTTP، وتم استخدام HTTP Cache Manager من أجل محاكاة سلوك التخزين المؤقت للمتصفح خلال فترة الاختبار، وتم استخدام CSV Date Set Config من أجل تمرير متغيرات إلى صفحة الويب أثناء الاختبار، على سبيل المثال صفحة منتج نحتاج في كل مرة تمرير ID الخاص به في عنوان الصفحة، وذلك حتى لا يتم تكرار نفس الصفحة في كل نشاط للمستخدم.



الشكل رقم (10) منهجية الاختبار بواسطة JMeter.

5.1.3. مرحلة تقرير النتائج والتوصيات: يتم في هذه المرحلة تجميع النتائج التي يتم الحصول من أدوات الاختبار ضمن PrefTester manager بهدف اصدار تقرير نهائي بنتائج السيناريوهات السابقة، والتوصيات اللازمة لتحسين جودة وأداء تطبيق الويب، يوضح الشكل رقم (11) خطوات جمع وتحليل نتائج الاختبار.



الشكل رقم (11) خطوات جمع وتحليل نتائج الاختبار.

بهدف الحصول على تقرير مفصل عن نتائج الاختبار، ومقارنة هذه النتائج مع المتطلبات التي تم جمعها في مرحلة جمع المتطلبات والتي تتضمن زمن الاستجابة المتوقع لكل صفحة ويب، تم اقتراح ضمن أداة إدارة عملية الاختبار ما يلي:

1- إدخال كافة نتائج المقاييس التي تم الحصول عليها من أداة GTmetrix مع التوصيات المقدمة لتحسين كفاءة وأداء صفحة الويب، تقدم تقريرا لكل صفحة يحتوي على المقاييس التالية:

- مؤشر أول عرض للمحتوى: هو مقياس أداء يقيس مدى سرعة مشاهدة الزوار للمحتوى الفعلي، (مثل النصوص والصور والفيديو وما إلى ذلك) ضمن صفحة الويب، وهو يمثل إجمالي الوقت المستغرق من بداية تحميل الصفحة إلى النقطة التي يتم فيها عرض أي محتوى على الشاشة.
- سرعة الصفحة: هو زمن يتم حسابه باستخدام الاختلافات المرئية بين الإطارات التي تم التقاطها أثناء تحميل الصفحة، يمثل كل إطار مرئي فترات زمنية، ويتم تسجيل كل إطار تم التقاطه من أجل الاكتمال

المرئي فوق الجزء المرئي من الصفحة حتى يصبح كل المحتوى مرئيًا، بمعنى اخر عندما تكون الصفحة مكتملة بصريًا.

- مؤشر أكبر محتوى: يقيس الوقت الذي يستغرقه أكبر "عنصر محتوى" (على سبيل المثال، صورة رئيسية .....) على صفحة الويب ليصبح مرئيًا في إطار عرض الزائر.
- الزمن اللازم للتفاعل: يقيس الوقت اللازم حتى تكون الصفحة جاهزة بشكل موثوق لتفاعل المستخدم.
- زمن حظر العرض: يقيس زمن حظر العرض إجمالي الوقت الذي تم فيه حظر صفحة الويب، بمعنى أخر يقيس الزمن الذي لا يسمح للمستخدم خلاله من التفاعل مع صفحة الويب.
- الازاحة في تخطيط صفحة الويب: هو مقياس أداء تم تقديمه في عام 2020 بواسطة Lighthouse لقياس الاستقرار المرئي الملحوظ أثناء تحميل الصفحة.(https://gtmetrix.com/)

2- إدخال كافة نتائج المقاييس التي تم الحصول عليها من أداة JMeter بهدف معرفة زمن الاستجابة لكل صفحة ويب في سويات الحمل المختلفة.

يقدم JMeter تقريرا يحتوي على عدة معايير إحصائية هامة لزمن الاستجابة وهي:

1. Min Response Time: وهو أقل زمن استجابة لصفحة الويب خلال سيناريو الاختبار
2. Max Response Time: وهو أعلى زمن استجابة لصفحة الويب خلال سيناريو الاختبار
3. Median Response Time: وهي القيمة التي تكون 50% من قيم زمن الاستجابة أصغر منها والنصف الأخر أكبر منها.
4. 90^{th} pct Response Time: وهي القيمة التي يكون 90% من قيم زمن الاستجابة أصغر منها.
5. 95^{th} pct Response Time: وهي القيمة التي يكون 95% من قيم زمن الاستجابة أصغر منها.
6. 99^{th} pct Response Time: وهي القيمة التي يكون 99% من قيم زمن الاستجابة أصغر منها.
7. Error%: وهي نسبة الخطأ في عدد الطلبات الفاشلة في صفحة الويب بالنسبة إلى طلبات الHTTP الموجودة ضمن صفحة الويب.
8. Transaction/s: وهي إنتاجية الصفحة خلال سيناريو الاختبار.

بهدف تحليل هذه النتائج، ومعرفة الصفحات التي تحتاج إلى إجراء تحسينات تم اقتراح حساب المقاييس التالية بعد إضافة كافة مقاييس زمن الاستجابة التي نحصل عليها في كل سيناريو إلى أداة إدارة عملية الاختبار وهي:

أ- قياس المتوسط الحسابي لقيم زمن الاستجابة.

ب- بما أن المتوسط الحسابي غير كافي لتقدير بأن زمن الاستجابة مثالي، حيث من الممكن أن يكون زمن الاستجابة للمستخدم الأول (2) ثانية وللمستخدم الخامس (4) ثانية وللمستخدم رقم 50 هي (8) ثانية يكون المتوسط (4.6) ثانية وهي قيمة ربما تكون مقبولة، لذلك تم اقتراح حساب الانحراف المعياري لكافة قيم زمن الاستجابة لكل صفحة ويب خلال سيناريو الاختبار، بهدف قياس مدى تشتت وتبعثر القيم عن قيمة المتوسط الحسابي وكلما كانت هذه القيمة أعلى كلما كانت الصفحة تملك أداء أسوأ.

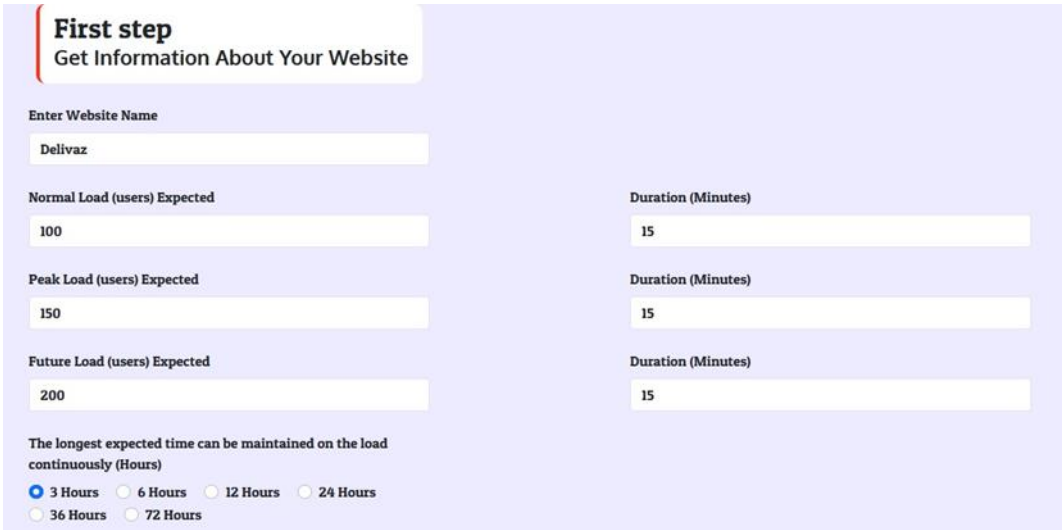
ج- بهدف مقارنة كافة قيم زمن الاستجابة التي نحصل عليها خلال سيناريو الاختبار مع القيمة المطلوبة لصفحة الويب تم اقتراح حساب مقياس جديد يسمى الانحراف عن المطلوب يعتمد هذا المقياس على حساب تشتت وتبعثر القيم عن القيمة المطلوبة بدلا من حساب التبعثر عن المتوسط الحسابي، وبالتالي كلما زاد هذا المقياس نستنتج أن هذه الصفحة تمتلك أداء أسوأ، وتخرج عن مستوى الخدمة المطلوب.

نلاحظ أن الإطار خلال هذه المرحلة يقدم نتائج تكون شاملة لكافة السيناريوهات التي يتعرض لها التطبيق، وبالتالي التنبؤ بسلوك وأداء التطبيق في كافة الحالات، حيث يحتوي التقرير على زمن الاستجابة المتوقع، والإنتاجية المتوقعة، ونسبة الأخطاء في تحميل طلبات الصفحة، وذلك خلال سويات الحمل الطبيعية، وفي أوقات الذروة، وفي ظروف الحمل المتوقعة في المستقبل، وخلال التغير المفاجئ في الحمل، وخلال أطول مدة زمنية ممكن أن يخضع خلالها التطبيق الى حمل مستمر، بالإضافة إلى تقييم أداء وبنية صفحة الويب بواسطة أداة GTmetrix وفق ما تم ذكره عن الية تقييم هذه الأداة لصفحة الويب في مرحلة التنفيذ ضمن الإطار المقترح، وهذه النتائج مجتمعة تساعد مطوري التطبيقات في اتخاذ القرارات المناسبة لتحسين الأداء.

2.3. اختبارات الأداء بالاعتماد على الإطار المقترح:

تم اختيار أحد مواقع التجارة الالكترونية وهو (https://www.delivaz.com) لإجراء اختبارات الأداء، وهو ما يزال في مرحلة التجريب واجراء الاختبارات اللازمة، وتكون النتائج أكثر دقة لأن الحمل المطبق على الموقع من أداة الاختبار فقط، سيقدم التطبيق خدمة توصيل طلبات من المطاعم والمتاجر للزبائن، وسيعمل في أكثر من دولة، حاليا يتضمن عدة متاجر ومطاعم بعد اختيار دولة أوكرانيا مدينة أوديسا، لذلك من الضروري إجراء اختبارات الأداء للتحقق من سلوك التطبيق في سويات مختلفة من الحمل، حيث تم بالمرحلة الأولى جمع المتطلبات اللازمة لعملية الاختبار وفق ما هو محدد في أداة إدارة عملية الاختبار وذلك خلال عدة مراحل موضحة بالأشكال التالية:

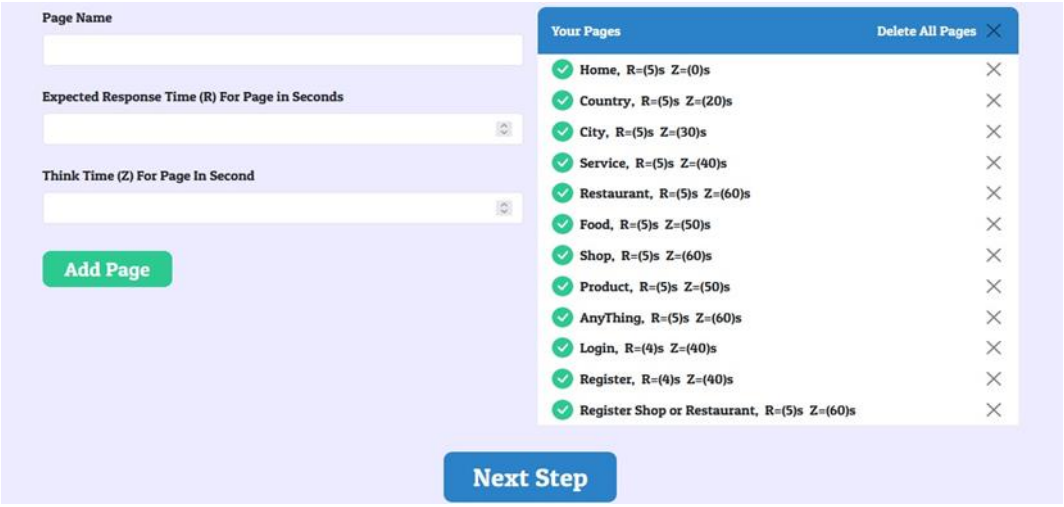
أ- الشكل رقم (a.12) جمع سويات الحمل المتوقعة من خلال PrefTester Manager وذلك بهدف اختبار التطبيق ضمن هذه السويات المختلفة، والتنبؤ باستجابة التطبيق في كافة الحالات.



الشكل رقم (a.12) جمع سويات الحمل المتوقعة من خلال PrefTester Manager

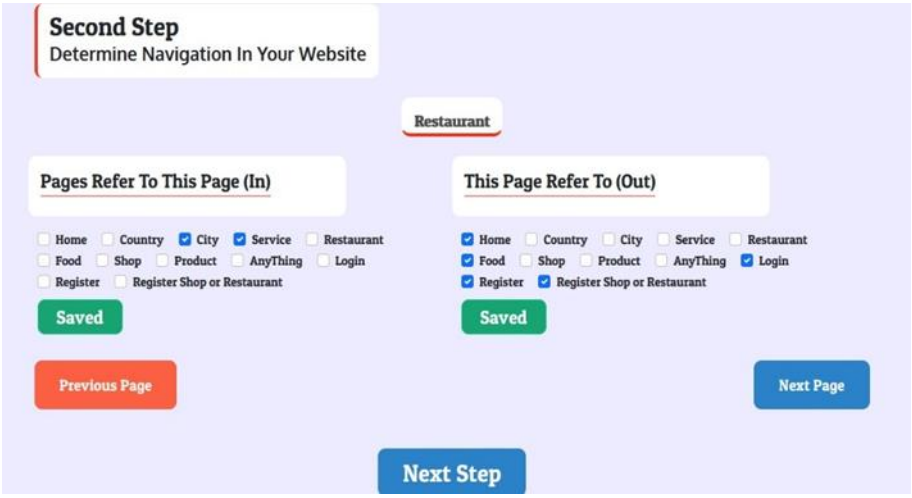
ب- الشكل رقم (b.12) جمع زمن الاستجابة وزمن التفكير المتوقع لكل صفحة من خلال PrefTester Manager وذلك بهدف مقارنة زمن الاستجابة المتوقع لكل صفحة مع الزمن الذي نحصل علية خلال سيناريوهات الاختبار المختلفة، وبالتالي معرفة إمكانية تحقيق التطبيق لاتفاقيات مستوى الخدمة المطلوب، ويتم تطبيق زمن التفكير لجعل المستخدم المولد من أداة الاختبار نموذجا عن المستخدم الحقيقي ينتظر بعض الوقت في صفحة الويب قبل الانتقال للصفحة التالية، ونلاحظ من خلال هذا الشكل قد تم جمع بيانات الصفحة الرئيسية، وصفحة الدولة (أوكرانيا)، وصفحة المدينة (أوديسا)، وصفحة الخدمات ضمن أوديسا، وصفحة المطعم، وصفحة المتجر، وصفحة الوجبة، وصفحة المنتج، وصفحة تسجيل الدخول،

وصفحة التسجيل، وصفحة التسجيل كمطعم أو كمتجر، وصفحة خدمة (anything OR Express Delivery).



الشكل رقم (b.12) جمع زمن الاستجابة والتفكير المتوقع لكل صفحة من خلال PrefTester Manager

ج- الشكل رقم (c.12) يوضح الروابط الداخلة والخارجة من صفحة المطعم، ويتم تحديد كافة الروابط لكل الصفحات السابقة بنفس الطريقة، من أجل معرفة أهمية الصفحات من خلال خوارزمية Page Rank، وتحديد إمكانية تجوال المستخدم في موقع الويب، من أجل بناء نشاطات محتملة للمستخدم على الموقع، وتتم هذه المرحلة من خلال PrefTester Manager.



الشكل رقم (c.12) الروابط الداخلة والخارجة الخاصة بصفحة المطعم من خلال PrefTester Manager

د- الشكل رقم (d.12) تكوين نشاطات متوقعة للمستخدم من خلال PrefTester Manager وذلك بهدف تطبيق هذه النشاطات على أداة الاختبار، نلاحظ من خلال هذا الشكل نشاطان محتملان للزائر يبدأ كل منها بالصفحة الرئيسية وينتهي بصفحة الوجبة، وتم تكوين نشاطات أخرى للزائر من خلال الصفحات الأخرى سيتم ذكرها في نموذج الحمل المقترح.

الشكل رقم (d.12) تكوين نشاطات متوقعة للمستخدم من خلال PrefTester Manager

هـ- الشكل رقم (e.12) تحديد متغيرات النشاطات من خلال PrefTester Manager وذلك بهدف معرفة إمكانية تكرار النشاط من قبل المستخدم، وتحديد Pacing time للتحكم في معدل إرسال وتنفيذ الطلبات على المخدم، ونلاحظ من خلال هذا الشكل مغيرات النشاطان الأول والثاني، كما تم تحديد هذه المتغيرات لكافة النشاطات المولدة.

الشكل رقم (e.12) تحديد متغيرات النشاطات من خلال PrefTester Manager

و- الشكل (13) نتائج تنفيذ خوارزمية Page Rank تساعد في توزيع سويات الحمل المختلفة بناء على أهمية الصفحة، وبالتالي بناء نموذج الحمل المقترح، ونلاحظ من خلال هذا الشكل اختلاف الأهمية بين صفحات الموقع.

Home : Rank is 10
Login : Rank is 10
Register : Rank is 10
Register Shop or Restaurant : Rank is 3.438
Country : Rank is 2.815
City : Rank is 1.361
Restaurant : Rank is 1.042
Shop : Rank is 1.042
Service : Rank is 1.035
AnyThing : Rank is 1.035
Food : Rank is 1
Product : Rank is 1

الشكل رقم (13) نتائج تنفيذ خوارزمية Page Rank

ثم في المرحلة الثانية بناء خطة الاختبار بواسطة أداة إدارة علمية الاختبار، موضحة بالأشكال التالية:

أ- الشكل رقم (a.14) هدف ومجال خطة الاختبار المحددة من خلال PrefTester Manager.

Fifth step
Get Performance Plane

Document	Performance Testing Plane For Simulate User Experience
Test ID	164407287232275594096
Website	Delivaz
Objective	Develop Performance test Plan for the Delivaz website to ensure if the current system architecture is optimal for handling increased traffic on the website and also, to figure out if there are any upgrades that need to be done.
Scope	1-Performing the user experience performance simulator without a specific load. 2-Performing the user experience performance simulator with a Expected Normal load (Load Testing). 3-Performing the user experience performance simulator with a Expected Peak load (Peak Load Testing). 4-Performing the user experience performance simulator with a Expected Future load (Capacity Testing). 5-Performing the user experience performance simulator with a Un Expected load (Spike Testing). 6-Performing the user experience performance simulator with a Expected load During a Long Time (Endurance Testing). 7-Propose suggestions

الشكل رقم (a.14) هدف ومجال خطة الاختبار المحددة من خلال PrefTester Manager

ب- الشكل رقم (b.14) أدوات وبيئة الاختبار المحددة من خلال PrefTester Manager.

Out of Scope	1-Functional or accuracy testing of the Delivaz website. 2-Any other testing types not included in the Scope section.
Hardware	Load Generator, Ex: (Dell Intel Core i7 -3.40 GHz., 16GB RAM).
Software	Operating System, EX: (Windows Or Linux Or Mac)
Performance Testing Tool	1-GTmetrix. 2-Apache-JMeter-5.4.1.
Environment	1-Java Runtime Environment (JRE) is installed to run JMeter. 2-A good internet connection is Confirmed.
Risks	The load JMeter applying is not the only load for the web site. Also, at the same time, there will be a website getting requests from the outside world. So, the report analysis might be not very correct. The Best Time To Execute This Plane After 11 PM.

الشكل رقم (b.14) أدوات وبيئة الاختبار المحددة من خلال PrefTester Manager

ج- الشكل رقم (c.14) سيناريوهات الاختبار المحددة من خلال PrefTester Manager، ونلاحظ أن السيناريو الأول من أجل اختبار الأداء بدون أية سوية حمل، والسيناريو الثاني لاختبار الحمل (الموافقة لسويات الحمل الطبيعية)، السيناريو الثالث لاختبار أقصى حمل (الموافقة لسويات حمل الذروة)، والسيناريو الرابع لاختبار السعة (الموافقة لسويات الحمل المستقبلية)، والسناريو الخامس لاختبار تغير الحمل المفاجئ،

والسيناريو السادس لاختبارات التحمل (أطول مدة زمنية ممكن أن يخضع التطبيق خلاله إلى حمل مستمر).

Performance Test Scenarios	Scenarios are explained in: 1-Scenario 1 For One User. 2-Scenario 2 For Load Testing. 3-Scenario 3 For Peak Load Testing. 4-Scenario 4 For CapacityTesting. 5-Scenario 5 For Spike Testing. 6-Scenario 6 For Endurance Testing.
Entry Criteria	Functionally stable application.
Exit Criteria:	All the Performance objectives are met.

الشكل رقم (c.14) سيناريوهات الاختبار المحددة من خلال PrefTester Manager

د- الشكل رقم (d.14) محددات سيناريو الاختبار الأول المحددة من خلال PrefTester Manager، والتي توضح المقاييس التي نحصل عليها بواسطة الأداة المختارة وهي GTmetrix، ومنهجية التنفيذ والنتائج المتوقعة من هذا السيناريو.

Scenario 1	Test user Experience with out load
Objective	Validate Response Time For All Pages in Delivaz By measuring: 1-First Contentful Paint. 2-Time to Interactive. 3-Speed Index. 4-Total Blocking Time. 5-Largest Contentful Paint. 6-Cumulative Layout Shift.
Performance Testing Tool	GTmetrix
Approach and Execution Strategy	1-Select Any Browser. 2-Go To https://gtmetrix.com/ 3-Enter URL For Target Page.
Results	Report For Every Page Contain ResponseTime Details.

الشكل رقم (d.14) محددات سيناريو الاختبار الأول المحددة من خلال PrefTester Manager

هـ- الشكل رقم (e.14) محددات سيناريو الاختبار الثاني المحددة من خلال PrefTester Manager.

Scenario 2	Test user Experience with Expected Normal Load
Objective	Validate Response Time For Every Page in Delivaz
Performance Testing Tool	Apache-JMeter-5.4.1
Duration	Ramp-Up period: 900 seconds
Approach and Execution Strategy	Apply Workload modelling For Load testing On JMeter
Results	Report Contain Throughput and Response Time For Every Page.

الشكل رقم (e.14) محددات سيناريو الاختبار الثاني المحددة من خلال PrefTester Manager

و- الشكل رقم (f.14) محددات سيناريو الاختبار الثالث المحددة من خلال PrefTester Manager.

Scenario 3	Test user Experience with Expected Peak Load
Objective	Validate Response Time For Every Page in Delivaz
Performance Testing Tool	Apache-JMeter-5.4.1
Duration	Ramp-Up period: 900 seconds
Approach and Execution Strategy	Apply Workload modelling For Peak testing On JMeter
Results	Report Contain Throughput and Response Time For Every Page.

الشكل رقم (f.14) محددات سيناريو الاختبار الثالث المحددة من خلال PrefTester Manager

ز- الشكل رقم (g.14) محددات سيناريو الاختبار الرابع من خلال PrefTester Manager.

Scenario 4	Test user Experience with Expected Future Load
Objective	Validate Response Time For Every Page in Delivaz
Performance Testing Tool	Apache-JMeter-5.4.1
Duration	Ramp-Up period: 900 seconds
Approach and Execution Strategy	Apply Workload modelling For Capacity testing On JMeter
Results	Report Contain Throughput and Response Time For Every Page.

الشكل رقم (g.14) محددات سيناريو الاختبار الرابع من خلال PrefTester Manager

ح- الشكل رقم (h.14) محددات سيناريو الاختبار الخامس المحدد من خلال PrefTester Manager.

Scenario 5	Test user Experience in Spike Testing
Objective	Validate Response Time For Every Page in Delivaz And Server Resource Monitoring
Performance Testing Tool	Apache-JMeter-5.4.1
Duration	Ramp-Up period: 100 seconds
Approach and Execution Strategy	Apply Workload modelling For Load testing On JMeter
Results	Report Contain Throughput and Response Time For Every Page.

الشكل رقم (h.14) محددات سيناريو الاختبار الخامس المحدد من خلال PrefTester Manager

ط- الشكل رقم (i.14) محددات سيناريو الاختبار السادس المحدد من خلال PrefTester Manager.

Scenario 6	Test user Experience with Expected Normal Load With Long Duration
Objective	Validate Response Time For Every Page in Delivaz And Server Resource Monitoring
Performance Testing Tool	Apache-JMeter-5.4.1
Duration	Ramp-Up period: 10800 seconds
Approach and Execution Strategy	Apply Workload modelling For Endurance testing On JMeter
Results	Report Contain Throughput and Response Time For Every Page.

الشكل رقم (i.14) محددات سيناريو الاختبار السادس المحدد من خلال PrefTester Manager

ي- الشكل رقم (a.15) محددات النشاطان الأول والثاني في نموذج الحمل المقترح والتي تم حسابها وفق ما تم ذكره سابقا في مرحلة تخطيط اختبار الأداء ضمن الإطار المقترح، وتتضمن معلومات عن تسلسل الصفحات ضمن النشاط، وعدد المستخدمين المتوقع لهذا النشاط في سيناريوهات الاختبار المتبقية (الثاني

والثالث والرابع والخامس والسادس)، ومتغيرات النشاط (زمن ال Pacing، عدد مرات التكرار)، والزمن المتوقع لإنهاء النشاط.

Workload modelling
NL: Number of Users in Load testing
NP: Number of Users in Peak Load testing
NC: Number of Users in Capacity testing
ND: Number of Users in Endurance testing
Z: Page Think Time (seconds)
P: Pacing (seconds)
L: Loop Count
TT: Time To End Iteration By One User (seconds)

Step	NL	NP	NC	ND	P	L	TT
1- Home Z= 0 2- Country Z= 20 3- City Z= 30 4- Service Z= 40 5- Restaurant Z= 60 6- Food Z= 50	3	5	6	37	60	2	290
1- Home Z= 0 2- Country Z= 20 3- City Z= 30 4- Restaurant Z= 60 5- Food Z= 50	3	5	7	41	60	2	245

الشكل رقم (a.15) محددات النشاطان الأول والثاني في نموذج الحمل المقترح

ك- الشكل رقم (b.15) محددات النشاطات الثالث والرابع والخامس في نموذج الحمل المقترح.

1- Home Z= 0 2- Country Z= 20 3- City Z= 30 4- Service Z= 40 5- Shop Z= 60 6- Product Z= 50	3	5	6	37	60	2	290
1- Home Z= 0 2- Country Z= 20 3- City Z= 30 4- Shop Z= 60 5- Product Z= 50	3	5	7	41	60	2	245
1- Home Z= 0 2- Country Z= 20 3- City Z= 30 4- AnyThing Z= 60	4	6	8	48	60	3	190

الشكل رقم (b.15) محددات النشاطات الثالث والرابع والخامس في نموذج الحمل المقترح

ل- الشكل رقم (c.15) محددات النشاطات السادس والسابع والثامن والتاسع والعاشر والحادي عشر في نموذج الحمل المقترح.

1- Home Z= 0 2- Country Z= 20 3- Register Z= 40	8	12	16	97	60	1	134
1- Home Z= 0 2- Login Z= 40 3- Register Z= 40	11	16	21	127	60	1	153
1- Home Z= 0 2- Register Z= 40	11	16	21	127	60	1	109
1- Home Z= 0 2- Country Z= 20 3- Login Z= 40	8	12	16	97	60	1	134
1- Home Z= 0 2- Login Z= 40	11	16	21	127	60	1	109
1- Home Z= 0 2- Register Z= 40 3- Login Z= 40	11	16	21	127	60	1	153

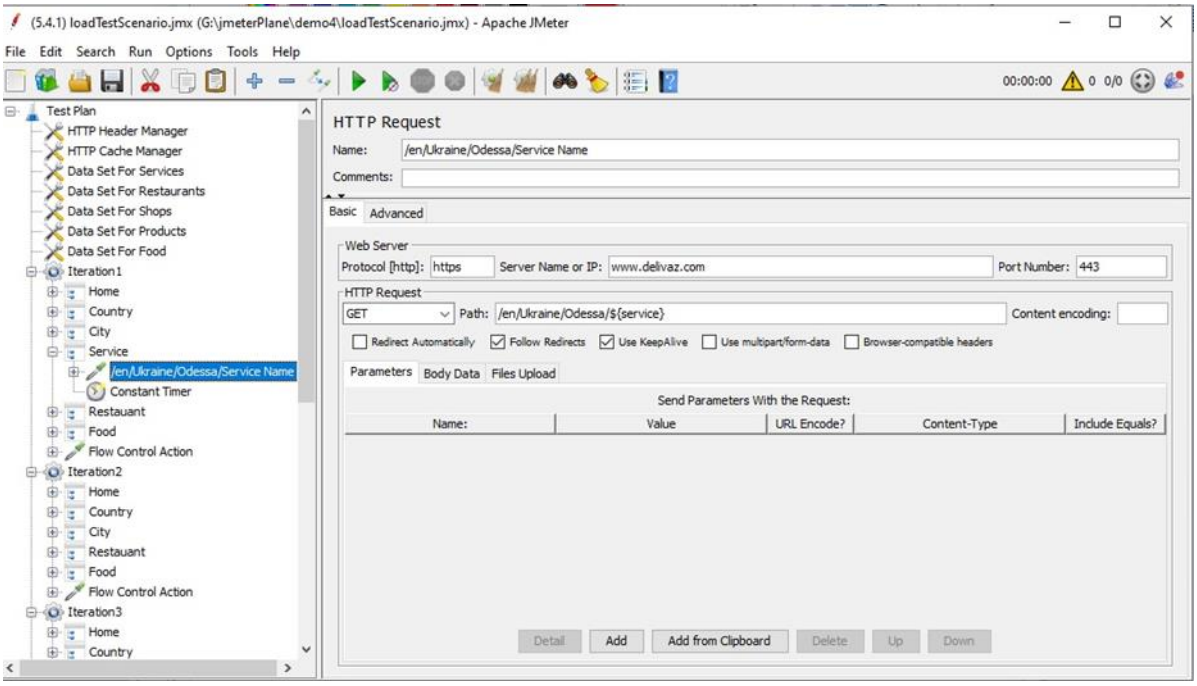
الشكل رقم (c.15) محددات النشاطات السادس والسابع والثامن والتاسع والعاشر والحادي عشر في نموذج الحمل المقترح

م- الشكل رقم (d.15) محددات النشاطات الثاني عشر والثالث عشر والرابع عشر والخامس عشر في نموذج الحمل المقترح.

1- Home Z= 0 2- Country Z= 20 3- City Z= 30 4- Service Z= 40 5- Restaurant Z= 60 6- Register Shop or Restaurant Z= 60	3	5	7	42	60	1	300
1- Home Z= 0 2- Country Z= 20 3- City Z= 30 4- Restaurant Z= 60 5- Register Shop or Restaurant Z= 60	4	6	8	47	60	1	255
1- Home Z= 0 2- Login Z= 40 3- Register Z= 40 4- Register Shop or Restaurant Z= 60	9	13	18	106	60	1	218
1- Home Z= 0 2- Register Z= 40 3- Register Shop or Restaurant Z= 60	8	12	17	99	60	1	174

الشكل رقم (d.15) محددات النشاطات الثاني عشر والثالث عشر والرابع عشر والخامس عشر في نموذج الحمل المقترح

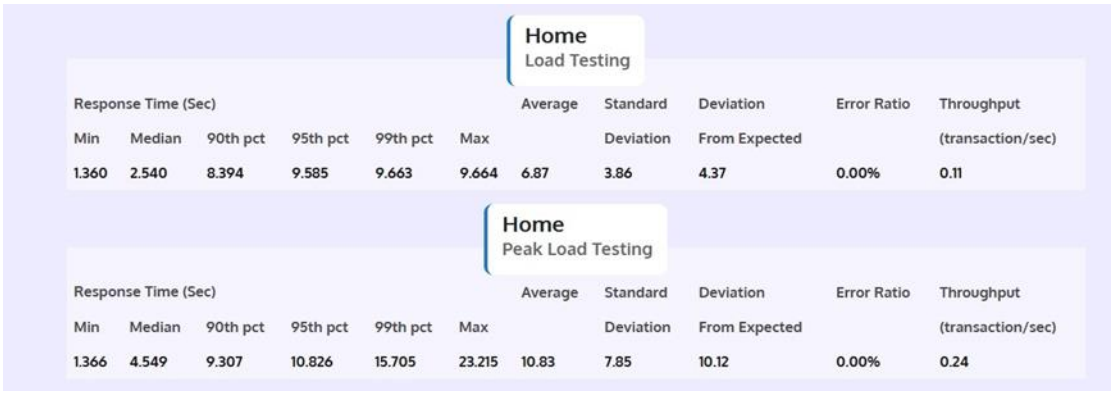
ثم تم بالمرحلة الثالثة والرابعة برمجة وتنفيذ اختبارات الأداء وفق ما تم ذكره سابقا ضمن الإطار المقترح، ويوضح الشكل رقم (16) سيناريو الاختبار بواسطة JMeter حيث يتم تنفيذ السيناريو كما هو وارد في خطة الاختبار.



الشكل رقم (16) سيناريو الاختبار بواسطة JMeter

ثم في المرحلة الخامسة تم ذكر نتائج الاختبار للصفحة الرئيسية فقط، لأن الهدف توضيح طريقة عرض وتحليل الإطار المقترح للنتائج، علما ان التقرير يحتوي النتائج لكافة الصفحات والتي يبلغ عددها 12 صفحة وبنفس النموذج، وبالتالي يمكن معرفة عيوب كل صفحة بشكل منفصل، حيث توضح الأشكال التالية نتائج الاختبار.

أ- الشكل رقم (a.17) نتائج اختبار الصفحة الرئيسية خلال السيناريو الثاني والثالث، ونلاحظ من خلال الشكل قيم مقاييس زمن الاستجابة التي سبق ذكرها في مرحلة تقرير النتائج والتوصيات من الإطار المقترح.



Home
Load Testing

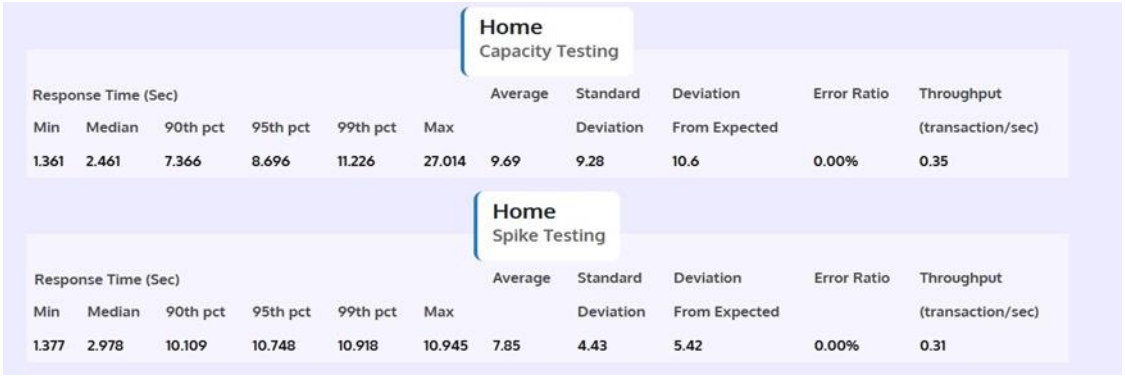
Response Time (Sec)						Average	Standard Deviation	Deviation From Expected	Error Ratio	Throughput (transaction/sec)
Min	Median	90th pct	95th pct	99th pct	Max					
1.360	2.540	8.394	9.585	9.663	9.664	6.87	3.86	4.37	0.00%	0.11

Home
Peak Load Testing

Response Time (Sec)						Average	Standard Deviation	Deviation From Expected	Error Ratio	Throughput (transaction/sec)
Min	Median	90th pct	95th pct	99th pct	Max					
1.366	4.549	9.307	10.826	15.705	23.215	10.83	7.85	10.12	0.00%	0.24

الشكل رقم (a.17) نتائج اختبار الصفحة الرئيسية خلال السيناريو الثاني والثالث

ب- الشكل رقم (b.17) نتائج اختبار الصفحة الرئيسية السيناريو الرابع والخامس.



Home
Capacity Testing

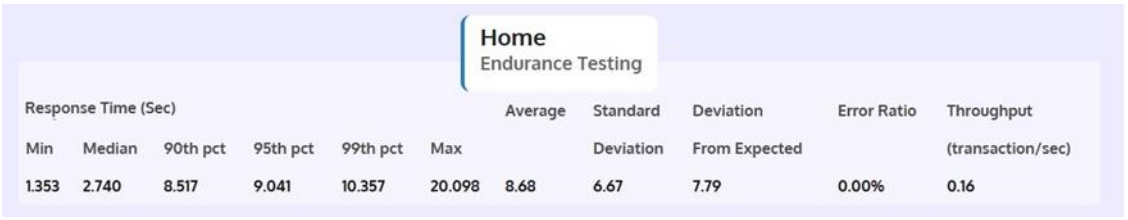
Response Time (Sec)						Average	Standard Deviation	Deviation From Expected	Error Ratio	Throughput (transaction/sec)
Min	Median	90th pct	95th pct	99th pct	Max					
1.361	2.461	7.366	8.696	11.226	27.014	9.69	9.28	10.6	0.00%	0.35

Home
Spike Testing

Response Time (Sec)						Average	Standard Deviation	Deviation From Expected	Error Ratio	Throughput (transaction/sec)
Min	Median	90th pct	95th pct	99th pct	Max					
1.377	2.978	10.109	10.748	10.918	10.945	7.85	4.43	5.42	0.00%	0.31

الشكل رقم (b.17) نتائج اختبار الصفحة الرئيسية السيناريو الرابع والخامس

ج- الشكل رقم (c.17) نتائج اختبار السيناريو السادس.



Home
Endurance Testing

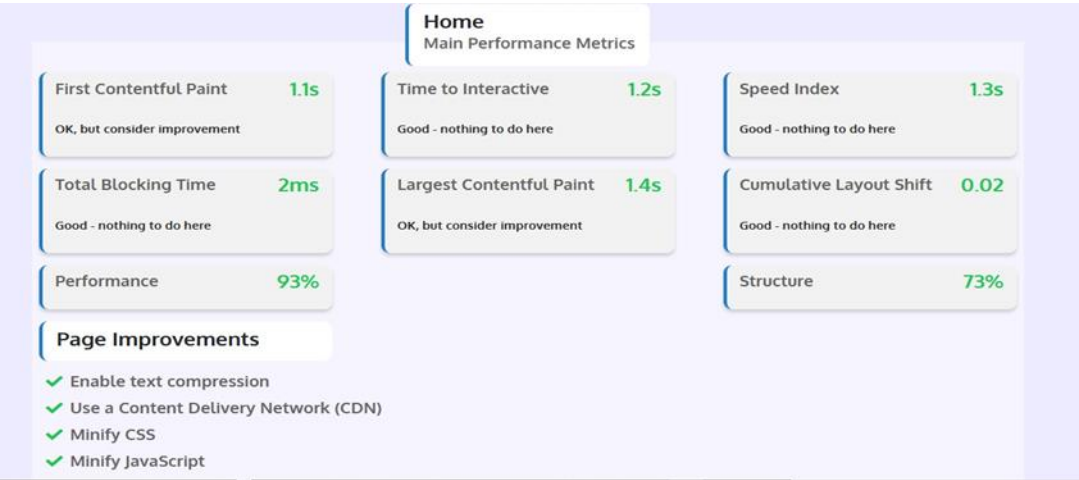
Response Time (Sec)						Average	Standard Deviation	Deviation From Expected	Error Ratio	Throughput (transaction/sec)
Min	Median	90th pct	95th pct	99th pct	Max					
1.353	2.740	8.517	9.041	10.357	20.098	8.68	6.67	7.79	0.00%	0.16

الشكل رقم (c.17) نتائج اختبار الصفحة الرئيسية السيناريو السادس

د- الشكل رقم (d.17) نتائج اختبار السيناريو الأول والتوصيات اللازمة لتحسين الأداء، والتي توضح أن أداء صفحة الويب ممتاز بدرجة 93%، وبنيتها بدرجة 73%، وهناك بعض التحسينات لجعل البنية أفضل متمثلة فيما يلي:

- ضغط النصوص.
- استخدام شبكة توصيل المحتوى.

- تصغير ملفات ال css.
- تصغير ملفات الjavaScrript .



الشكل رقم (d.17) نتائج اختبار السيناريو الأول والتوصيات اللازمة لتحسين الأداء

حيث أن النتائج في كل من السيناريوهات الثاني وحتى السادس، والتي تم تنفيذها بواسطة JMeter بهدف التنبؤ بزمن الاستجابة والإنتاجية ونسبة الخطأ في تحميل طلبات صفحة الويب، وذلك ضمن سويات الحمل المختلفة التي تحقق أنواع اختبارات الأداء كاملة، ونلاحظ في مثالنا هذا أن زمن الاستجابة يزداد في ظروف حمل الذروة، حيث كان متوسط زمن الاستجابة هو (10.83 Sec) وكذلك مؤشر الانحراف عن المطلوب سجل أعلى قيمة (10.12)، بينما في ظروف الحمل الطبيعية سجل مؤشر الانحراف عن المطلوب أدنى قيمة (4.37) وسجل متوسط زمن الاستجابة (6.87 Sec) ، وهي قريبة عن القيمة المطلوبة والمحددة في مرحلة جمع المتطلبات.

أما نتائج الاختبار في السيناريو الأول المنفذ بواسطة GTMetrix يتضمن تفاصيل زمن الاستجابة لكل صفحة ويب مع ذكر كافة التحسينات الممكنة لتحسين أداء الصفحة، ويتم تقديم هذا التقرير لمطوري التطبيق لأنها ستشكل مرجعا لهم من أجل تحسين الأداء العام للتطبيق وتجنب الوقوع في مشاكل الأداء مستقبلا.

يمكن الإشارة إلى أنه لو طبق الإطار المقترح على موقع اخر، وبما أن الإطار يلبي جميع مراحل الاختبار كما ورد في فقرة الإطار المقترح، فإن المختبر سوف يدخل قيم جديدة لسويات الحمل ويدخل صفحات الموقع، ثم سيتم تحليل الروابط بين هذه الصفحات وإنتاج خطة اختبار شاملة لكافة السيناريوهات ولكن متوافقة مع قيم سويات الحمل الجديدة وأهمية صفحات الموقع، ثم يتم اتباع المنهجيات الخاصة بكل أداة اختبار في مرحلة البرمجة والتنفيذ، ثم يتم إدخال النتائج إلى أداة إدارة عملية الاختبار من أجل تحليلها ومقارنتها مع متطلبات حالة الاختبار الجديدة، وهذا يراعي التعقيد في طبيعة تطبيقات الويب، حيث أن لكل صفحة من صفحات الويب زمن استجابة متخلف يعتمد على حجمها والوظائف التي تقدمها، كما أن زمن التفكير يختلف من صفحة لأخرى، أيضا سويات الحمل المتوقعة تختلف باختلاف التطبيق.

4.مناقشة النتائج.

بعد مراجعة الدراسات السابقة الواردة المبحث الأول، نلاحظ أن جميع المنهجيات المتبعة في عملية اختبار الأداء لا تلبي معظم مراحل ومفاهيم الاختبار الواردة ضمن المبحث الأول، ومن عيوبها أنه لم يتم جمع متطلبات كافية لبناء خطة اختبار تحاكي كافة أنواع الاختبارات المطلوبة، وإنما كانت معتمدة على بعض ميزات الأداة المستخدمة، حيث أن كل المنهجيات المنفذة من خلال أداة JMeter كانت تعتمد تطبيق سوية أو سويتين فقط من

الحمل على صفحة الويب، من دون مراعاة أن المستخدم في البيئة الواقعية يقوم بزيارة عدة صفحات متتالية من الموقع، بمعنى اخر يجب في البداية دراسة الموقع وتحليل النشاطات الممكنة للمستخدم على الموقع بهدف الاعتماد عليها في بناء خطة الاختبار، ومنه نستنتج أن سلوك المستخدم المولد بواسطة أداة الاختبار لم يكن قريبا من سلوك المستخدم الحقيقي، وبالتالي عدم القدرة على اكتشاف كافة المشاكل والثغرات المتعلقة بالأداء والتي يتعرض لها المستخدم وخاصة خلال ظروف مختلفة من الحمل، وعدم التحقق من قدرة التطبيق على تلبية معايير جودة الأداء واتفاقيات مستوى الخدمة المطلوبة.

أما في الإطار السابق الذي ورد ضمن الدراسات السابقة، والذي تضمن اختبار أداء عدة مواقع لم يتم ذكرها حفاظا على خصوصيتها، ولكن المنهجية المتبعة ضمن هذا الإطار لا تقدم تصورا عن أداء هذه المواقع في سويات مختلفة من الحمل، ولم يتم ذكر خطة للاختبار علما أنها من أساسيات مراحل الاختبار، وإنما تم تحديد زمن الاستجابة فقط كمعيار لتحديد الموقع الأفضل.

أما بعد تصميم الإطار المقترح PrefTester والذي يجمع ميزات كل من أداتي الاختبار JMeter & GTMtrix حيث يتم باستخدام GTmetrix تنفيذ السيناريو الأول بهدف اختبار الأداء والبنية لصفحة الويب وتقديم التوصيات اللازمة لتقليل زمن الاستجابة إلى أقصى حد ممكن، كما يتم باستخدام JMeter تنفيذ بقية السيناريوهات الأخرى والتي تتضمن كافة سويات الحمل التي قد يخضع لها موقع الويب، بهدف التنبؤ بأداء التطبيق خلال السويات المختلفة من الحمل، كما تم بناء أداة لإدارة عملية الاختبار PrefTester Manager بهدف جمع متطلبات شاملة عن موقع الويب تضمن بناء خطة اختبار تجعل من سلوك المستخدم في بيئة الاختبار أقرب ما يمكن إلى سلوكه في البيئة الواقعية، وتوفر الميزات التالية:

أ- جمع سويات الحمل المتوقعة.

ب- تحديد متطلبات جودة الخدمة، حيث يتم تسجيل زمن الاستجابة المتوقع لكل صفحة من صفحات التطبيق بهدف مقارنتها مع النتائج التي نحصل عليها في السويات المختلفة.

ج- تحديد النشاطات الممكنة للمستخدم على الموقع بعد إدخال كافة الروابط الداخلة والخارجة لصفحة الويب.

د- حساب أهمية الصفحات ضمن الموقع حيث أن الحمل لا يكون متساوي على جميع الصفحات، وإنما يوجد صفحات ذات أهمية أكثر وممكن أن تكون بوابة للعبور من خلالها للصفحات الأخرى، وهي التي تحدث اختناقات في أداء الويب.

هـ- بناء خطة اختبار شاملة يمكن تحميلها كملف PDF أو تبقى مخزنة ضمنن قاعدة البيانات الخاصة بها تحتوي كافة المعلومات الضرورية واللازمة لأجراء اختبارات الأداء المطلوبة.

و- بناء نموذج الحمل المتوقع ضمن خطة الاختبار، يتضمن هذا النموذج كافة النشاطات المرشحة من قبل المختبر، والمعلومات اللازمة لجعل سيناريو الاختبار المنفذ بواسطة أداة JMeter أقرب ما يمكن للواقع، حيث يحوي على عدد المستخدمين المتوقع في كافة سيناريوهات الاختبار الواردة ضمن الخطة، وعدد مرات التكرار للنشاط خلال مدة الاختبار، والزمن المتوقع لإنهاء النشاط من قبل المستخدم.

ز- تخزين نتائج الاختبار التي نحصل عليه من أداتي الاختبار، وحساب المتوسط الحسابي والانحراف المعياري والانحراف عن القيمة المطلوبة لزمن الاستجابة لكل صفحة من صفحات التطبيق.

بمعنى اخر يقدم الإطار المقترح لكل مرحلة من مراحل الاختبار منهجية، ولكل نوع اختبار يوجد سيناريو مقترح ضمن الخطة، وتم اختيار أداتين مجانيتين لهما ميزات مختلفة من أجل مرحلة برمجة وتنفيذ الاختبار، وبناء

أداة لإدارة عملية الاختبار تلبي المراحل الأخرى من مراحل الاختبار (مرحلة جمع وتحليل المتطلبات، مرحلة التخطيط، مرحلة تقرير النتائج والتوصيات)، واقتراح منهجية خاصة لأداة ال JMeter تختلف عن المنهجيات المقترحة في الأبحاث السابقة التي اعتمدت على هذه الأداة، وبالتالي نلاحظ أن التكامل بين هذه الأدوات JMeter & GTMtrix & PrefTester Manger يحقق كافة المفاهيم والمراحل اللازمة للتحقق من أداء تطبيق ما والتي وردت في الدراسة النظرية من هذا البحث.

5.الأعمال المستقبلية.

يمكن تطوير الإطار المقترح ليشمل منهجيات اختبار تطبيقات الهاتف الجوال من خلال البحث المنهج عن الأدوات المجانية القادرة على تلبية متطلبات الاختبار الخاصة بها، واقتراح المنهجيات لتطوير خطة الاختبار اللازمة لتنفيذها، وذلك لأن طبيعة بناء تطبيقات الهاتف الجوال تختلف عن تطبيقات الويب.

قائمة المراجع.

- Matam, S., & Jain, J. (2017). Pro Apache JMeter: web application performance testing. Apress.
- Shivakumar, S. K. (2020). Modern Web Performance Optimization. Springer Books.
- Rodrigues, A. G., Demion, B., & Mouawad, P. (2019). Master Apache JMeter-From Load Testing to DevOps: Master performance testing with JMeter. Packt Publishing Ltd.
- Hayek, M., Farhat, P., Yamout, Y., Ghorra, C., & Haraty, R. A. (2019, September). Web 2.0 Testing Tools: A Compendium. In 2019 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT) (pp. 1-6). IEEE.
- Srivastava, N., Kumar, U., & Singh, O. (2021). Software and Performance Testing Tools. Journal of Informatics Electrical and Electronics Engineering, 2(1), 1-12.
- Molyneaux, I. (2014). The art of application performance testing: from strategy to tools. " O'Reilly Media, Inc.".
- Pradeep, S., & Sharma, Y. K. (2019, February). A pragmatic evaluation of stress and performance testing technologies for web based applications. In 2019 Amity International Conference on Artificial Intelligence (AICAI) (pp. 399-403). IEEE.
- Sasankar, P. (2019). Analysis of Test Management, Functional and Load Testing Tools.
- Hayek, M., Farhat, P., Yamout, Y., Ghorra, C., & Haraty, R. A. (2019, September). Web 2.0 Testing Tools: A Compendium. In 2019 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT) (pp. 1-6). IEEE.
- Wang, J., & Wu, J. (2019, June). Research on Performance Automation Testing Technology Based on JMeter. In 2019 International Conference on Robots & Intelligent System (ICRIS) (pp. 55-58). IEEE.
- Maila-Maila, F., Intriago-Pazmiño, M., & Ibarra-Fiallo, J. (2019, April). Evaluation of Open Source Software for Testing Performance of Web Applications. In World Conference on Information Systems and Technologies (pp. 75-82). Springer, Cham

- Hidayanto, R., & Sawitri, P. Performance Testing of e-Payment Website Using JMeter.
- Jha, N., & Popli, R. (2017). Comparative Analysis of Web Applications using JMeter. International Journal of Advanced Research in Computer Science, 8(3).
- Kyrychenko, I., & Semko, D. (2021). PERFORMANCE TESTING, ITS USABILITY AND TECHNIQUES. InterConf.
- Hossain, M. T., Hassan, R., Amjad, M., & Rahman, M. A. (2021). Web Performance Analysis: An Empirical Analysis of E-Commerce Sites in Bangladesh.
- Suryadevara, S., & Ali, S. (2020, June). Preperformance Testing of A Website. In CS & IT Conference Proceedings (Vol. 10, No. 7). CS & IT Conference Proceedings.
- Putri, M. A., Hadi, H. N., & Ramdani, F. (2017, November). Performance testing analysis on web application: Study case student admission web system. In 2017 international conference on sustainable information engineering and technology (SIET) (pp. 1-5). IEEE.
- Joshi, S., & Goel, S. (2020). Comparative Study of Page Rank and Weighted Page Rank Algorithm. Available at SSRN 3919359.
- https://jmeter.apache.org/usermanual/get-started.html.
- https://gtmetrix.com/blog/everything-you-need-to-know-about-the-new-gtmetrix-report-powered-by-lighthouse/#grading-scoring-details.