DECISION MAKING DASHBOARD FOR AGILE TEST AUTOMATION DEVELOPMENT

M.M.S.U.MAHAGEDARA

198763L

Dissertation submitted to the Faculty of Information Technology, University of Moratuwa, Sri Lanka for the partial fulfillment of the requirements of the Honours Degree of Master of Science in Information Technology.

JULY 2022

Declaration

I declare that this is my own work and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

The above candidate is carrying out research for the Master of Information Technology Dissertation under my supervision.

UOM Verified Signature

Signature of the Student Date

M.M.S.U.Mahagedara

Signature of the Supervisor Date

Dr.Chaman Wijesiriwardane

Dedication

This thesis is wholeheartedly dedicated to my loving parents Lal Mahagedara and Sajeewani Karunarathna Mahagedara for their endless love, support and encouragement throughout my journey. They have been my constant source of inspiration and given me the drive and discipline to tackle tasks with enthusiast and determinations. Without their patience, understanding, support and most of all love this work would not have been made possible.

Acknowledgement

I would like to convey my gratitude towards Dr.Chaman Wijesiriwardana for supervising this project and giving valuable feedback on the implementing the application.

And also I thank all the participants who took the survey and who supported to confirm this application's importance and to understand need of the proposed application in the software industry.

Last but not least, I would like to thank my loving parents for supporting and encouraging me to achieve all the goals and appreciate all the endless efforts and dedication they have done for me in my journey.

Abstract

Automatic testing enables faster iterations and reliable test outputs which leads to the delivery of a high-quality product. Test Automation in Agile development improves efficiency of development and faster execution, Reusability of test component on Repetitive nature of tasks, Higher test coverage, and helps to reduce overall testing cost to provide a high quality product. In Agile development environment, developers and quality engineer's work together and in every sprint and releases set of features to the product decided by product owners. Quality engineers in agile team covers manual testing and test automation for the features within the sprint allocated. In most of the companies agile team tracks and display manual testing progress, bugs and defects found, test coverage, test cases for the new features developing but not the test automation progress and results. In some of the startups and even in the multinational level companies, in their product teams, quality engineers doesn't have any proper mechanism to provide test automation results in proper way. Most of the quality engineers has to execute the test automation scripts manually on the local machine and give sprint updates and test automation progress updates on based on the console results. It's hard to explain and show the results and test automation coverage on console for the offshore product owners and the team. As a solution this application will help project managers, business analysts and offshore product owners and the whole team to make decisions based on the test results within the sprint. This proposed application will help quality engineers to update the completed test automation results on the dashboard within the sprint so management and the agile team will be able to make decisions whenever necessary. And quality engineers does not need to execute in local machine and give updates to the management and the team as it's already updated on proposed dashboard. This proposed application would function as a decision making dashboard using test automation results for the small scale agile projects.

Keywords: Test Automation, Decision Making Dashboard, Test Results, Project management, Agile

Table of Content

List of Abbreviations

Abbreviation	Description
ITes	Information Technology Enabled Services
POM	Page Object Model
QA	Quality Assurance
HIPAA	the Health Insurance Portability and Accountability Act
CI / CD	Continuous integration and continuous delivery/continuous deployment
IDE	Integrated development environment
HTML	Hyper Text Markup Language
API	Application Programming Interface
DTO	Data Transfer Object
PHI	Protected Health Information
CRUD	Create, Read, Update, Delete

List of Figures

Chapter 1

Test Automation in Agile Environment

1. Introduction

This chapter describes the background of the research area followed by the background and literature survey for decision making dashboard using agile test automation for the professional services ITes. The gap between the existing systems/processes and the proposed system along with the problem that paved the way for this research is discussed in the later part of this chapter.

To deliver a high quality product to the stakeholder and to the user, company should make sure that developing product meets it business requirements and customer satisfaction. For that every company do testing before delivering a product or adding a new feature to a product to ensure that its meeting client requirement and produces expected results. Testing is essential part of product development especially to ensure quality of developing product. Many smaller startups and mid-sized companies don't give testing a much attention even if it's essential for delivering a strong quality product. Just like it's important to have tests in place it's also matters how we run these tests. Some companies prefers to do manual testing, although that's not the best approach for fast moving developing product team. Therefore next logical step would be automating the testing process when we have the tests in place. Manual testing should be minimized at all because it requires high manual intervention and lot of human resources in a team and its time consuming which will expect human errors on complex lengthier test scenarios. And test automation escalations overall software development efficiency and allows for more robust products to be developed.

Agile is fast moving philosophy in software industry yet test automation was not introduced at early days. However, today agile testing is an essential part to deliver agile deliverables. In agile development and testing is happening parallel following a collaborative team work. Agile focuses on continuous delivery of working software to the

clients. To achieve the goals within two weeks of timeline agile product team has to work collaboratively to complete and release the functions and applications. Specially to all the developers, support team, DB team, DevOps and Quality Assurance engineers has to work on faster collaboratively while frequently interactive so that all are in the same page.

Test automation in agile environment offers faster feedback cycles and delivers faster validation on development stages in developing product. Test automation especially useful because it helps to detect problems or defects early on the development life cycle which will help to increase team efficiency. Automating testing procedure team has to spend less time validating newly developed features as well as already developed and see the impact on other already developed test scenarios. It also improves communication with other teams like product owners, support team, UI/UX team who reply on the results of these tests. By automating scenarios it will help to reduce business expenses as fewer resources are spent on the product. Automation allows to spend time writing new tests and add it up to test suites so it will increase the test coverage. And also automated testing allows developers to write more in depth complex test scenarios and use cases. Lengthy tests that are often avoided during manual testing and also those can be left without executing. Moreover, reusability of test cases are improved as most of the test automation projects uses frameworks which can reuse the test components in different test scenarios. POM model is a one of the framework use for designing test automation projects.

Test automation in a developing product in agile mode, reduce cost and resources needed and gives higher test coverage to the test scenarios and reusable components in test components and better insights on application as well as shows performance, data tables, file contents. And also automation helps to quickly determine stability of the product, improved accuracy and enables to find failure patterns. And also implementation and testing logs and test execution data is considered a great source of information for identify issues and defects in developing product, those are rarely used to get the whole picture of the testing or the development life cycle. But data gathered during the testing process specially test execution data can be a valuable source of information. This data

contains, status of the software, results: pass or fail, timing information i.e. when a piece of code was executed, details about the identity of the software or module that was executed. [2]

The data we get from one test automation cycle, can be used to extremely useful insights about the product or the feature under development and about the product as a whole. Data like test automation status, timeline, features automated, pass and fail rates, failure patterns, deployment issues identified, number of executed test automation cycles from test automation may help product owners and business analysts to make decisions about the current development and the concerns over the product. With that knowledge they can decide improvements to take over in next sprints and prioritize product backlog. The outmoded way of doing this has been for the tester to download the test result data in a excel sheet and run some analytical measures to find the patterns. This is often very unwieldy and un-intuitive, so developers and testers rarely end up completing and get use of it. This may results in missed prospects of finding risky and potentially disastrous patterns. [2]

Most of the leading USA based IT companies in Sri Lanka is still following less productive ways to collect test automation results to measure test automation progress in sprint wise and quarter wise. Almost every software company no matter the size of the company, they don't focus and make it important of test automation execution details. Most of the multi-national companies even tend to use open source fee tools to develop software. And most of the clients won't be willing to purchase any commercial tools to support agile development team and help product team. Therefore in test automation, most of the companies won't purchase any tools to support test automation or get teat automation execution data and use in analyzing the developing product features. In the software industry as Quality engineers it is our responsibility to assure the developing product is bug free product which meets user requirements. Therefore apart from developing test scripts it is our responsibility to produce test coverage to clients, management and product team. Normally we had to execute test automation scripts for the sprint and quarter then show the console results. Then explain about the features and the automation status of it to the head of quality engineering and to the management at

every end of quarter which is very time consuming and less in transparency. And it shows lack of professionalism at work place. Moreover, at the end of every sprint, in sprint end meetings quality engineers required to explain about test automation results of sprint features to development team and to the off shore product team and product owners without producing any proof. This is not a very productive way of giving updates to the agile team and to the off shore product owners and their team. The information provided by the decision making dashboards can be tremendously valuable for a quality engineer or a software developer working on a part of software application. A trend of continuous failures over a time period may indicate that the development team has consistently failed to fix a specific defect. This, in turn, could be a worthy trigger for the quality engineer to raise a red flag to the management team regarding the developing product. [2]

Historically, Quality assurance during Agile-driven projects faced numerous encounters. One challenge is the absolute number of components and features that must be tested and validate within a short time period. Another challenge is achieving 100% functional testing for each feature release. Hostile timelines make frequent, often monotonous and redundant scenario testing difficult to complete on time. [10]

To cater that, it is important to have a customized real time decision making dashboard for a product team to analyze test automation details within the sprint to make decisions about sprint items developing and the issues identified. And also it will help product team to cater data breaching issues as well as develop their own dashboard according to their preferences catering their needs.

Chapter 02

Test Automation in Agile Environment – Limitations

2. Research Problem

1. Introduction

This chapter discuss the research problem comprehensively with issues and limitations in agile product teams in respect to test automation execution details. Moreover, later part of this chapter questions about how much software industry make important the execution data and the usage of those data to analyze the developing product.

2. Research problem

An operational decision making dashboard for agile test automation development needs to show sprint test automation results with statistics in each sprint to help agile product team and offshore product team to make decisions on the developing product based on the test automation results which gives data to analyze failure rate, failure patterns, performance statistics, deployment issues identified within the sprint feature development. As explained in previous chapter, test automation execution data is crucial in a software developing product. After a successful test automation execution we get important data related to the developing software feature and yet software industry won't make those data much important for a product team. Therefore there is a high risk of missing defects in a developing feature and going live on production. As quality engineers in software industry working with international clients, quality engineers have high responsibility of assuring the quality of the product and the verifying that all the features are meeting its requirements as per to the client. Therefore quality engineers run manual and automated tests in developing feature. With test automation quality engineers allows to create test cases which executes automatically and it will help agile team to verify the new developing features exhaustively within a sprint many times before

delivering to production. And also test automation helps quality engineers to test existing features in the product without any hassle.

In reality, even though its agile methodology agile team may has to face many difficulties running a sprint and delivering a successful quality product features to end users. In a team, there are different individuals we may have to deal with and with all the team members has to work towards to a one goal to achieve. But in reality team members like product owners, scrum masters, quality engineering leads senior quality engineers may face complications and managerial issues with follower team members to work for a one goal. That may happen when newly joined team members aren't open up to the team even in standups mentioning their technical difficulties and asking for a technical support from a team. And also when some individuals won't contribute to the team with taking accountability for their tasks whole team would end up with complications at the sprint end. In reality, we may have to work with individuals who won't take accountability for the assigned task and even at the daily standups they won't mention the test automation script development and progress of it. Some Quality engineers mention that they are working on the test automation scripts in each and every daily standups verbally but at the end they might have not even touched it. And non-technical people in the project like product owners, project managers has to accept yet they can't check the code and understand it. At the sprint end meeting when incomplete automation scripts will make whole Quality Assurance team disappointed and make whole team at risk. And in agile, it is considered as completed task if only feature is developed, validated and completed test automation and fixed all the bugs. But missing one step completing test automation would spill over the feature in to the sprint and considered as not completed. Then the whole team has to work extra hours on next sprint covering up the test automation and if any defects found from the test automation to fix it. Therefore this kind of situation and individuals will drag whole team down and in reality most of the product owners, project managers and leads cannot handle this situation since team is following agile not micro management. Thus, it is better to have a real time decision making dashboard for the agile product team to keep track of teat automation progress for a sprint. So then not only technical side members' even non-technical people in the product team could check the progress of the team and if they are not progressing enough support them with technical

support to achieve the tasks. It is better to have a dashboard which shows sprint details and the automation scripts developing so all the team members are at the same page about the developing software product. But unfortunately, in the software market there is no any application which supports agile teams to keep track and check the test automation script developing and execution progress in real time. Hence, there is lacking an application which supports agile team to monitor test automation script developing and execution in real time to avoid above mentioned issues.

Moreover, in industry projects quality engineers responsibility to produce test coverage of a particular sprint to development team and offshore product owner and team to ensure the features releasing are bug free and are up to the par. But inappropriately, since almost every company using open source tools and software quality engineers has no way to present test automation details in a sprint end meeting for the product team. So in reality quality engineers are executing test automation scripts in their local machines and keep the execution results with them. Some companies uses testing framework reporting tools like testing reporting tool, Extent reporting tool, Cucumber reporting tool which provides HTML reports in their local machines. Then quality engineers collect all the html reports and zip it and mail it to clients or respective parties who refers to it. It is a hassle to work on collecting html reports for each and every test and most importantly it does not shows the professionalism in IT companies. While some companies using testing framework reporting plugin to generate HTML reports other companies, surprisingly even multi-national companies won't use at least these reporting plugins to their test automation projects. So quality engineers has to execute and show honesty to the team about the test execution pass rates and test coverage. Most importantly this would lead to slip some serious defects we may find in the developing application. In general, having no any method or proper application to produce test automation script's progress and execution data in an agile team to monitor test automation script development and execution details to analyze developing product and find defects get from periodic test script execution is a disastrous for developing product.

In multi-national companies and startups quality engineers has to produce test automation execution details and test coverage for the quarter. So that was presented on the quality

engineers local machine by executing in the local machine and show the console execution details which is not very presentable and not professional. And quality engineers has to schedule a meeting with head of Quality Assurance and execute test scripts in front of them and wait till execution finishes. Then quality engineers has to show the console to give idea about pass fail rates for the past quarter features. I found it very uncomfortable and unprofessional as technical individuals in the industry. So if there were a real time decision making dashboard customized for test automation script development and execution integrated with Jira management to get sprint details easily whoever refers to the dashboard would get an idea about current sprint, past sprints and quarter wise details of test execution. But in the software market there is lacking an application which offers real time updates and a decision making dashboard which aligns with project management tools to support sprint wise and get data to support agile teams.

Test automation provides essential statistics for developing features and more important areas to be tested not only in UI but also loading issues and failure patterns due to network traffic and feature performances in peak and off peak hours, deployment issues, and broken builds which means poor code quality and defects in the developing product. If we have a real time dashboard with test execution data we can host these same scripts and execute those in a grid and get execution details periodically so we can get how developing application functions in different time. So if quality engineers has no any method to produce the test automation execution details and make use of it to get better insights from it to develop a quality product, it shows how quality engineers hassle to show case these data and get use out of it and how other members has ignored these valuable data.

There are number of dashboards available on the market but almost every dashboard available is not open source which will add cost on third party products and will increase its products when license renewing. Most of the tools in the commercial market costs highly which will be over the budget and even they purchased the license renewing is also would be higher. If company started to use any third party commercial tool to their software product team has to depend on it and over the years budget would get raised. Even we purchased most of the time product team’s won’t get exactly what they are

expecting therefore most of the product teams in the industry are avoiding purchasing commercial tools on developing a software product. Clients are not interested on purchasing some commercial tools to develop another application therefore software companies are more tend to use open source tools to develop the applications. Hence, there is lacking an application which will provide these features as well as support financially to the project budget and use in the developing product

In healthcare industry most of the applications are built very strictly and concern about its production data. In these companies it has its own software development subsidiaries which will develop their own applications to use in their health care industry. Health care partners are very concern about PHI data as it could leads to some serious violance. For an instance, USA health care industry and partners are following HIPAA policy to protect PHI data as if employee could find a details of USA celebrity as a patient of a hospital in USA and employee could access to patient's private details such as NIC, full name, address, payment details, disease they suffering from and treatments they are having, from which hospital which is prohibited to access and disclosure to outside. Therefore international clients are very concern about even the software we are using to develop the products since all the data rotating are very private and confidential. And due to the HIPAA compliance issues some companies restrict using third party solutions to their products to avoid data breaching. When it comes to the software testing it is identified as even from a test automation project execution there could be a data breaching and login to certain products unauthorized manner. Therefore clients and management has to approve even solutions product team is using open source tools and it is encourage to develop own applications assuring its security. For that reason, applications designed for that kind of companies who is adhering to HIPAA compliances and PHI data breaching there is lacking an application to display test automation details to product team since product team is not encouraged to purchase any third party commercial tools.

Furthermore, there is no any software application to get test automation execution details connected to project management tool. If quality engineers has that kind of customized decision making dashboard, it will be a hassle free comfortable place to work with and manage sprint features and complete within due period. And quality engineers no need to

make extra effort on executing test scripts on their local machines and explaining each and every time product owners and management request as the results will be updated real time on proposed decision making dashboard for an active sprint in the agile management board. Product owners and business analysts in the agile team will be able to check the dashboard and take an idea about the current sprint development and performance is and helps to refine the product backlog for the future sprint planning based on the findings of test automation results. And also developers and support team also would benefit from this kind of application as they could find failure patterns and defects in the developing product within the sprint and fic and complete those within the sprint.

3. Summary

As a summary, software industry especially in agile environment there are unseen issues in reality which drags actual product being developed and its issues we could identify from the test automation execution. Therefore to support agile product team and make them to be more productive and deliver the product on time and produce a quality product there is lacking an application to get better perceptions of an applications which produces from test automation execution. Moreover, software industry works for different domains but as industry professionals we should always protect client data PHI and adhering to HIPAA compliances. To support all those mentioned issues there is lacking an application in the current market place developed specially for test automation which can get better insights.

Chapter 03

Test Automation in Agile Environment – Research Gap

3. Research Gap

1. Introduction

This chapter deliberate about the research gap for the above research area and will discuss exhaustively the issues and the gap needed to be filled in software applications to support agile projects and get use of test automation details in agile projects to get better views in developing product and make the product more quality and useful to the end user. Additionally, rest of the sections will compare applications in the market and commercial tools methodically to find out the need of this proposed decision making dashboard application.

2. Research gap

Although there are numerous dashboard plugins and reporting tools available on the market, most of the applications does not provide useful insights to the product team to identify the defects and issues in developing product and increase its quality. Most of the applications in the market providing free trials for a specific time period and which is costly for a small scale project to purchase initially and also will be over the budget when renewing the licenses.

In software industry, for test automation most of the companies uses reporting tools to display their execution results. Even in multi-national companies are using open source testing framework HTML reporting tools like TestNG reporting tool, Cucumber Reporting tool, Extent Reporting tool etc. The disadvantage is all the execution reports will be in the Quality Assurance engineers local machines and engineer has to send it either mail each individual HTML reports. This is very unprofessional way to communicate test execution details and we are missing most of the test execution data

from it. If we have a dashboard we could have identify failure patterns, peak off peak hours traffic issues, loading issues in specific feature level and defects in the features. But HTML file in local machine won't do so.

And companies which is following HIPAA compliances and having PHI data avoid using third party applications and plugins to avoid data breaching. Since clients with sensitive data are concerned with data breaching most of these companies cannot purchase any applications from the market but develop themselves to cater the issue. Therefore there is a lacking an application which will help to track test execution data with PHI in a secure way.

Commercial tools which are trust worthy will cost more to company and cannot purchase a sole monitoring dashboard itself. This thesis is focused on small scale companies, and for small scale companies or startups cannot afford commercial tools to use in test automation. So that is one gap having in small scale companies.

And most of the project management tools provide dashboards only to track defects and bugs ratio found in a specific sprint not automation test results within a sprint. The applications in the market will host automation scripts in their server and execute those in a grid and if one fails it will show with the error. This will not support agile team to find internal product team would have. For an instance, newly joined Quality Assurance engineers will start test automation at the middle of the sprint but may not communicate with the team in the daily stand up. Some may have technical difficulties to continue or irresponsibility could lead the whole product team to drag one feature. If test automation is not completed within a sprint it may have to spill over the feature to next sprint. Which means product team has to put extra effort to complete these missed tasks. To avoid that if there is a real time dashboard for the product team that could have been avoided.

Moreover, to identify those issues and completeness of the test automation for a sprint it is hard to monitor for non-technical people like product owner or business analysts or scrum masters. There is lacking an application which shows sprint wise test execution and its progress which can easily read by non-technical people.

Leading IT companies mostly execute test automation scripts in CI/CD pipeline and take the results as a bulk which will not give useful data for a sprint decision making. Applications in the market offers companies to host their scripts in their server and execute in periodically. But that doesn't allows to have PHI and HIPAA compliances as well as to support sprint workflows.

Moreover, there are no any dashboards to show sprint details connected to JIRA and collects all the team members test execution for a particular sprint and display in a real time dashboard. So then technical persons and non-technical persons in the agile product team would manage sprint work in daily standups and client meeting to produce the sprint wise progress of the product.

There is lacking an application which will help quality engineers to update test automation results within sprint and help agile development team and offshore product owners to make decisions on current sprint and refining future sprint backlog to provide high quality product to the consumers.

If a tool provides all that is needed and is relatively easy to use, use the tool. However, some tools that provide powerful means of measurement can be costly or hard to use. So you need to decide between purchasing a powerful tool and using an open source dashboard that may not be as powerful. Another consideration is how well the dashboard integrates with your development environment. Whether you buy or build a tool, consider what quality aspects should be shown and what frequency that they are measured. Does the dashboard perform quality-related tests when initiated by a user, showing results of tests executed during build or integration, or is it monitoring the production system. [18] Therefore there a lack of an application which suits for a small scale company when focusing on all the facts mentioned above to provide a means to make test automation data accessible and visible to the team and make decisions on it.

3. Summary

As a summary, this chapter explains about the research gap of this proposed research domain. According to the above discussed points it is proven that there is lacking an application which will provide sprint details connected to JIRA and collects all the

product team Quality Assurance engineers test automation execution results of a sprint in to one real time dashboard. As most of the test automation dashboards in the market provides dashboards to show defects in the test execution by hosting test scripts committed to a source control version like git or bitbucket. There for there is a need of real time dashboard which connects with JIRA to get sprint details and collects all the test execution data from team in to one decision making dashboard to get better insights of developing product.

Chapter 04

Test Automation in Agile Environment – Survey

4. Survey on Target Audience

1. Introduction

This chapter discuss about the survey ran out on target audience to prove the need of real time dashboard for an agile product team to solve internal team issues discussed in above chapters. It was essential to have a survey to bring out the existence and importance of a real time decision making dashboard especially for agile environment to support agile team to manage their work and find defect early and get better useful insights from it.

2. Survey

Decision making dashboard is important to every developing product team to get useful views on the developing product and make changes according to the findings. As discussed in previous chapters it is proved that even in proven successful methodology has unseen issues when it comes to the real world project team management. In software industry projects inside agile environment there could be less visibility of test automation script development within the team. Since non-technical people like product owners and project managers cannot read test automation scripts cannot monitor team member's progress over the sprint goals. And also quality engineers has hard time explaining sprint test automation script development progress to the agile product team and explains the test automation execution and its findings including defects and issues of the developing product. And also lack of an application which management, offshore team and even agile team could refer and plan the rest of the sprint and product backlog using test execution and test progress of a team. With compared to the other existing applications in the market the features each provide are different from each other and not exactly team wanted to cater the unseen real problems within the team. To find out how the multi-

national companies and startup companies handle these problems and how this kind of real time decision making dashboard would help their agile team to support to overcome the above mentioned issues I ran out a simple survey to simply understand how this proposed application going to support.

To see whether the importance of the proposed real time decision making application, I ran a survey on target audience who works on software industry for multinational companies and for startups. The survey is created using google forms. And send it out to my colleagues and former colleagues who are working on multi-national companies and small scale startups. This survey is targeted only on software industry individuals so I can assure they have solid understanding of agile environment and how test automation is important to a developing product. Survey is made on people who are worked on software industry in different companies who worked on agile software methodologies. Within the limited time I received 28 responses from the receivers. According to the survey results all the individuals participated are working on software industry over a period of time. And they have mentioned about their job role so we can get a clear idea about what each agile team members thoughts on the proposed application. Most of the responses were captured from product owners, Project managers, senior developers, senior Quality Assurance engineers, Lead business analysts, quality engineers, and developers.

71.4% audience have worked on the software industry more than 5 years so we can assume that they have more knowledge on agile product teams and importance of test automation execution data for a developing product and also about the process so they can decide the appropriateness of this proposed application to get use in their agile product teams. According to the captured responses 100% of participants agreed that decision making automation dashboard for an agile product team would be useful for a product team in agile environment.

And to find out which type of companies applicable to have the proposed application I have asked some questions from target audience and majority of them were from small scale companies and 35.7% who doesn't have a real time decision making dashboard

specifically for test automation execution and rest were from multinational companies some have used different dashboards for test automation in their projects.

Moreover, to discover if there were any decision making dashboard focused on test automation in software industry I have added some questions throughout the survey. And as a result 70.4% were participants mentioned that there were no any decision making dashboards focusing on test automation in their companies which will support to monitor test automation progress and get execution details align with sprint details. According to the participant's opinion, 100% of the participants agreed that if there were any decision making dashboard focused on test automation that would help to the team and decisions on the developing product in agile environment.

Lastly all the participants has mentioned who will be benefit from this kind of proposed real time decision making dashboard application and majority of them have mentioned agile team members will benefit from this kind of application to get more use of test automation data and get more idea about developing product. According to the participants this kind of real time decision making dashboard will be benefits to the quality engineers and decision makers in the product team.

Survey URL: https://forms.gle/1FSuU9QqCAJV1MZu9

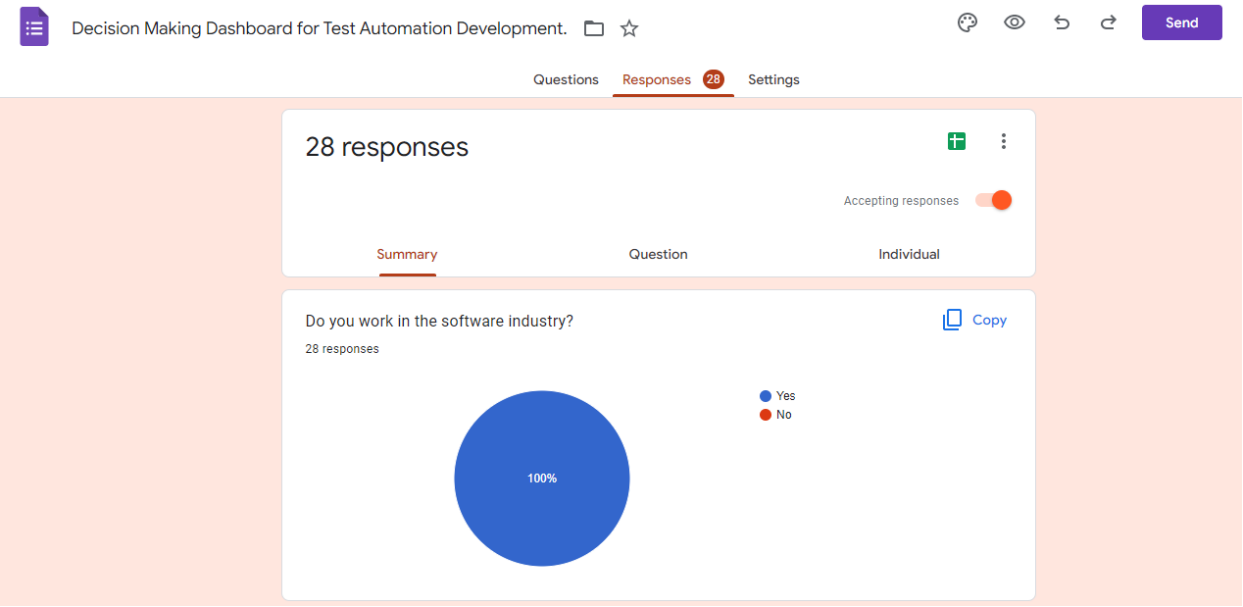

Figure 4.1: Survey responses 01

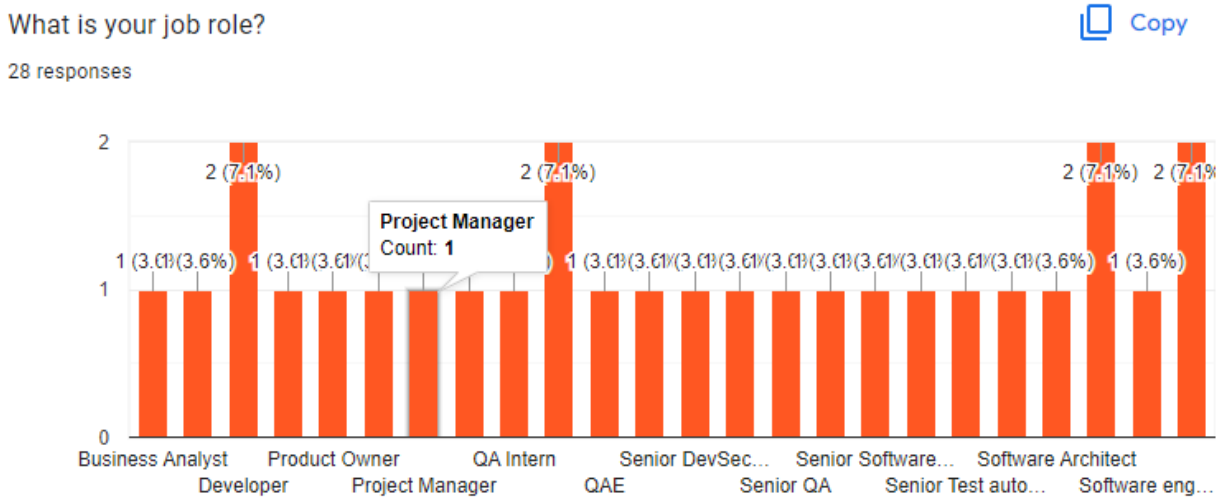

Figure 4.2: Survey responses 02

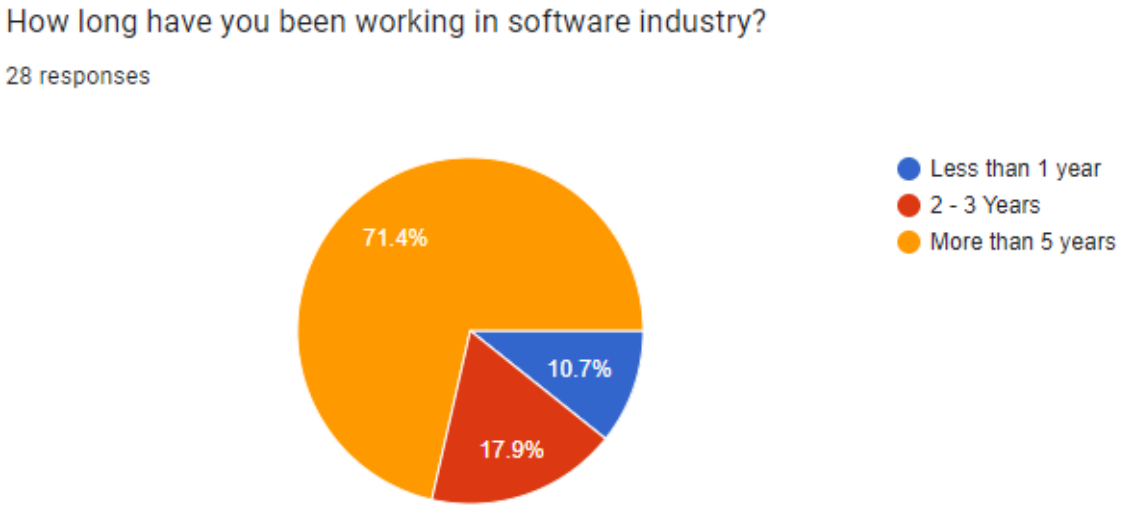

Figure 4.3: Survey responses 03

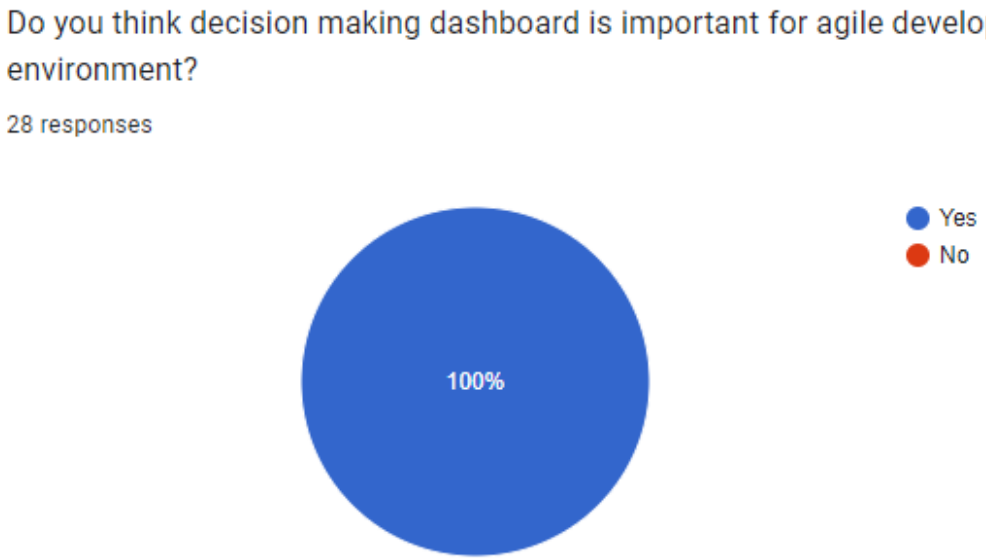

Figure 4.4: Survey responses 04

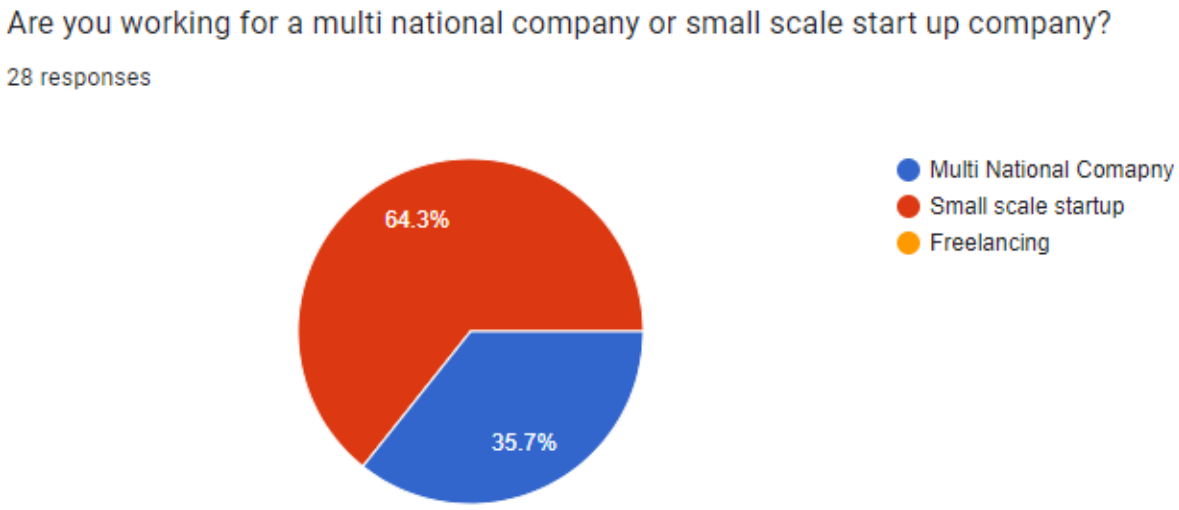

Figure 4.5: Survey responses 05

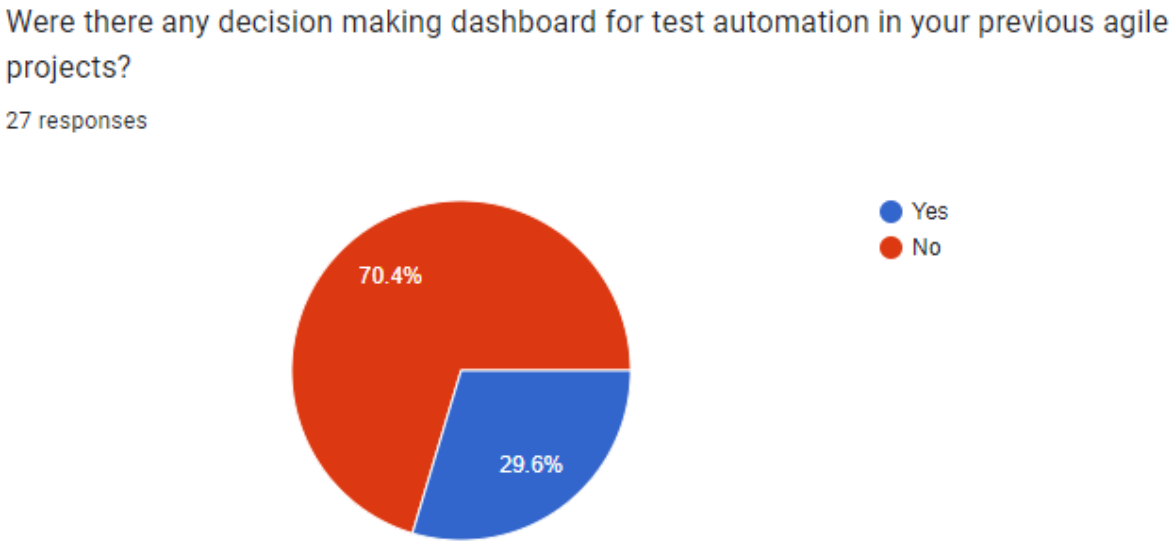

Figure 4.6: Survey responses 06

Figure 4.7: Survey responses 07

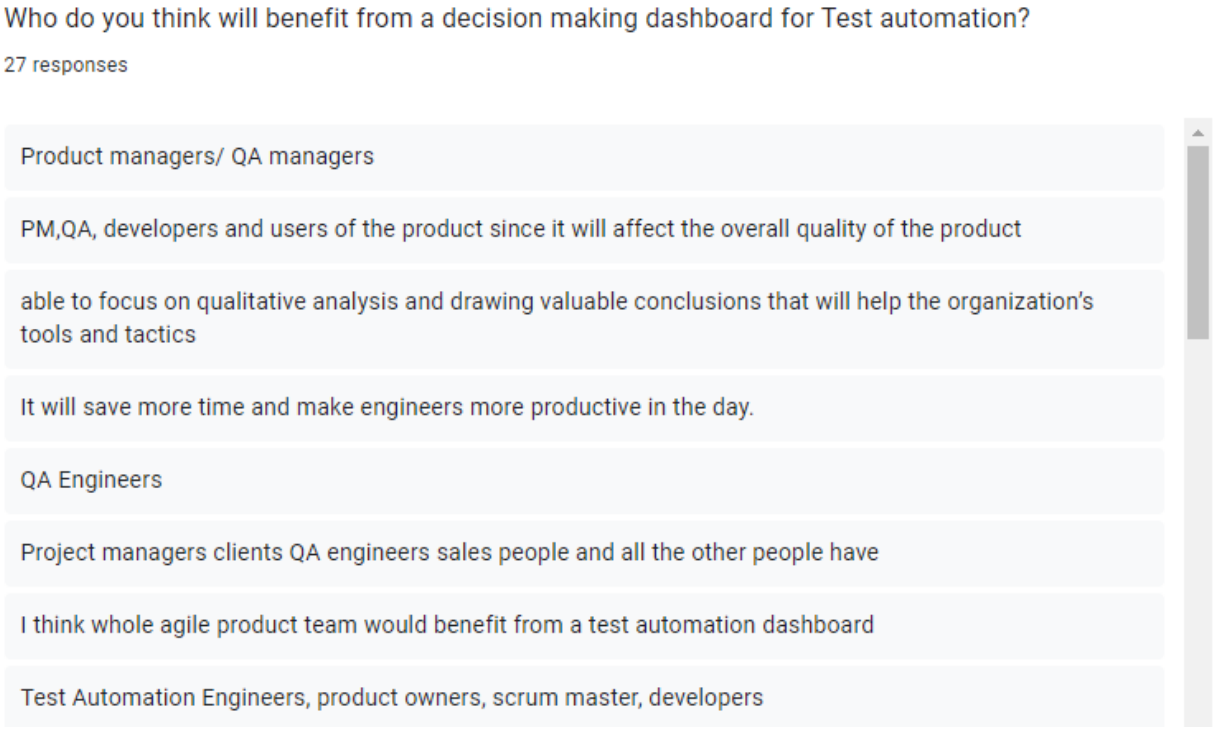

Figure 4.8: Survey responses 08

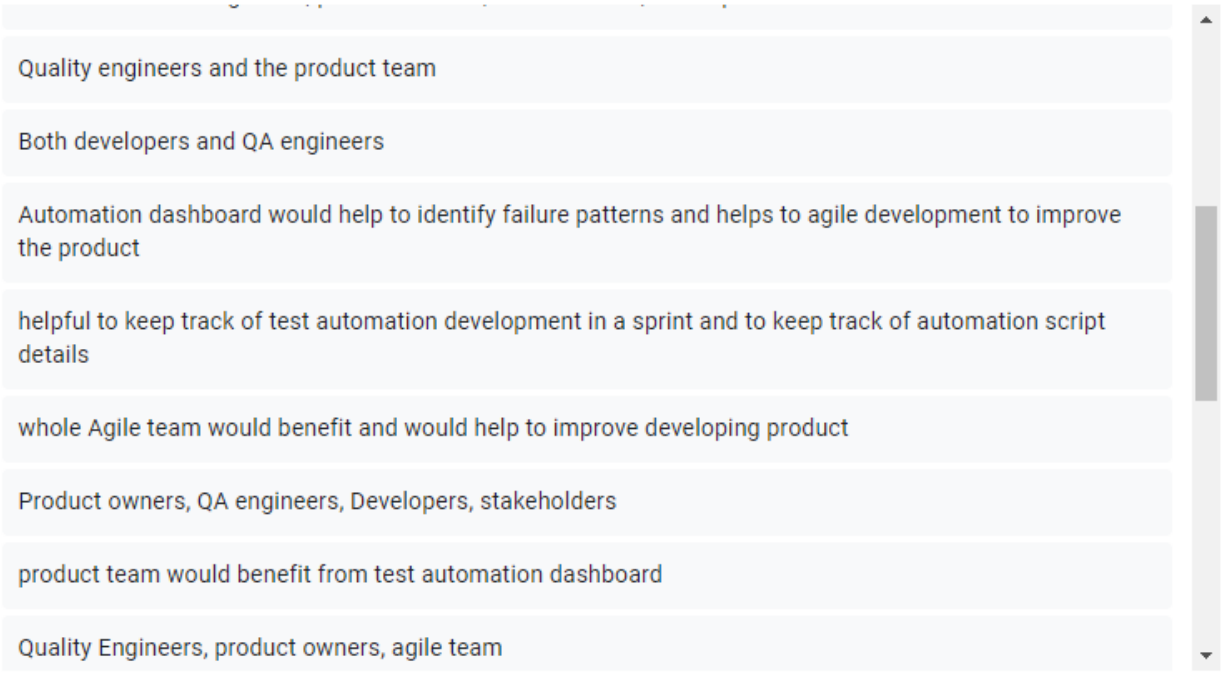

Figure 4.9: Survey responses 09

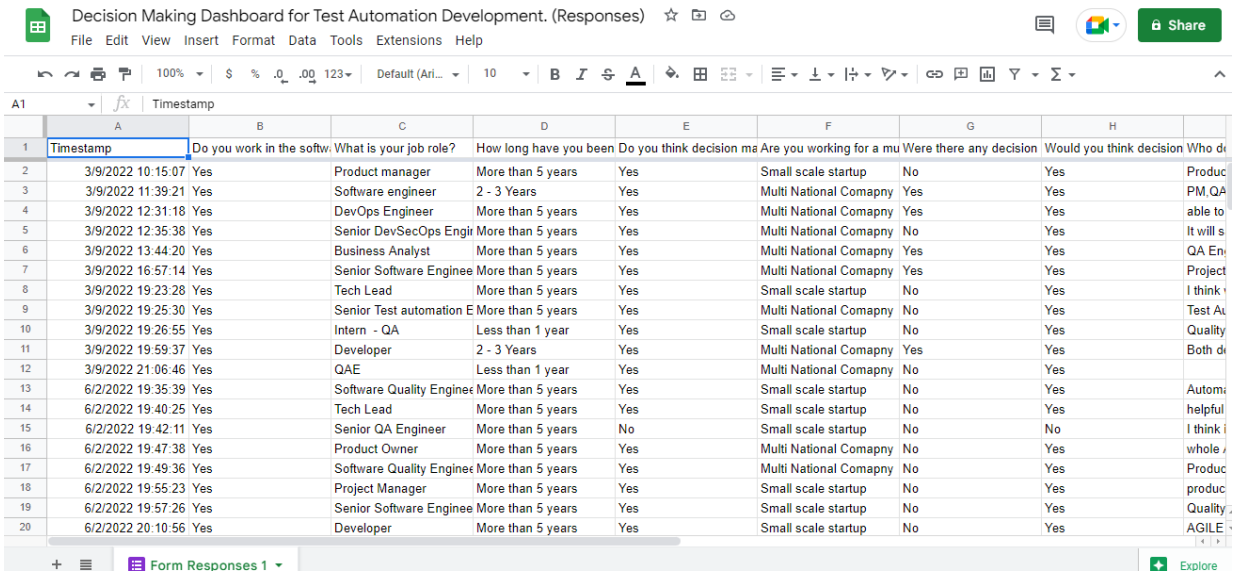

Figure 4.10: Survey responses 10

3. Summary

Based on the survey results, it is approved that the importance of implementing a real time decision making dashboard focusing on agile test automation is beneficial for agile product team as well as to develop product to ensure its quality within the short time of period. Subsequently the real time decision making dashboard will benefit agile team to keep track of test automation script development progress and execution and make better decisions on developing product and its features. And it is proved that majority of the companies mostly the small scale startup companies does not have any kind of dashboard specially focused on test automation. Moreover, the participants have stated that having a dashboard for test automation would support to improve the developing product in many ways.

Chapter 05

Test Automation in Agile Environment – Literature Survey

5. Background and Literature Survey

1. Introduction

In this chapter background, the review of researches that have been conducted and existing applications and their strengths and weaknesses are discussed comprehensively to support the proposed applications neediness. In this chapter I have referred to IEEE researches done by some researchers who are interested on agile and test automation execution and the different approaches to overcome different problems they have faced.

2. Background and literature survey

According to the research done by Loke Mun Sei, test reporting – test execution detailed log can be get using robot framework plugin which collects and publishes robot framework test results in Jenkins. With the robot framework plugin, it will produce a summary report file which contains overview of test execution results in HTML format including statistics based on tags, executed test suite and a log file contains details about detailed status on each of the test step whether it is failed or passed. [3]

Based on the study done by Dorothy and mark stated that the test results they got were unreadable to the non-technical person and by management. For that they have improved their systems to generate plots based on Crow-AMSAA model for reliability measurements and plugin to install to eclipse IDE so quality engineers can get test automation results. Silk Test from segue project has generated excessively large logs which is difficult to handle for that they have developed separate functions to handle each important test scenario.[4]

There are lot of tools and plugins available on the market but none of these provide a feature to support agile development team to monitor sprint test automation results to get useful insights. Calliope.pro is a one popular product available on the market which is a devops tool to display test results by hosting test scripts and sharing but can share only 25 test results in free trial version. [6] And it keeps historical data for 14 days which is not allowed in companies which is following HIPAA compliance policy. And this product is not focusing on capturing sprint wise details as usual only on uploaded test script execution and displays its results. [7]

Perfecto.io is another platform which provides test reporting and failure analysis for manual defect tracking. And also it monitors test automation scripts which is hosted and that will be provided only for paid subscriptions only. And will generate heap maps to high level idea of the test execution results. It is not focused on generating test automation results on sprint wise to support agile teams to monitor sprint progress. [8]

Report portal.io is a product which is focused on tracking defects and reporting all the automation results in one place again with paid subscription. And it allows to submit and track defects with full evidence (logs, screenshots, comments, etc.) directly from a failed test case into a bug tracking system. [9] All the product features supports for bulk test execution and won't support agile development to make useful data.

ZapTest is a latest platform which can use in test automation projects. It helps to execute test scripts on cross platform. Unfortunately it's a commercial tool with high budget. And won't be support real time dashboard update to support agile team.

Katalon studio is another platform uses for test automation. It is integrated with CI/CD pipelines and functioning in cloud and supports with visual and functional tests and powered big data and supports to make intelligent tests. This application helps to track automated and manual test results from central dashboards in testrail another third party application. This application enables continuous testing and do pre and post actions using external third party tools using customized extensions and plugins. Most importantly it will host scripts in cloud and execute those and will give a feedback to the team about the execution status. And this tool provides only a free trial and also have to purchase a

software and license renewing cost is also higher. This will be over budget for a small scale or startup companies to continue with their test automation projects.

To overcome challenges which Quality Assurance engineer are facing Incepts has developed a utility using selenium which uses Perl scripting to integrate the utility with database and web applications. After executing the automation scripts, the driver and the script produces the output folder for each and every execution with date and timestamp, which contains of pass or fail status for each of the data inputs to the tests. The output folder also has got screenshots. [10]

Above test reporting mechanism has focused on creating test reports but not making useful insights of it. They are lacking a decision making dashboard which helps agile team and management to get useful insights of it and take better decisions to deliver quality products to the stakeholders.

The research done by Hector I Dura shows that the importance of a dashboard in a development team. It is hard to realize, let alone advance, the quality of software without the knowledge of its software development process and the software products. There should be some dimensions and process to forecast the life-cycle of software products. The research described in this thesis is an attempt to analyses multiple metrics in Software Development production and show them in a Dashboard. So it provides a summary interpretation on Software Engineering, Software Quality and Software Metrics in a actual software corporation. The major results of this metrics, will be monitored in an easy to read, single page, real-time user interface, showing a graphical presentation of the current status and historical trends of the software production key performance indicators to enable instantaneous and informed decisions to be made at a glance. Metrics is a necessary factor. It is part of the necessary overhead for delivering the product. If team can detect emerging delivering risks early, then team can deal with them. The researched application is developed using SQL database, jobs and widgets to update the dashboard. The Jobs provide the data to the widgets. To specify which widget has to be used, is necessary assign the widget id. Dashing uses scheduler to schedule jobs. This scheduled job will execute every minute, and will send a random number to all widgets that have id. Jobs are where to put stuff such as fetching metrics from a database, or calling a third

party API. In the meantime the data fetch is happening in only one place, it means that all cases of widgets are in sync. Server Sent Events are used in order to stream data to the dashboards. [12]

3. Summary

As a summary this chapter discusses about past studies done on test automation execution data which researchers attempts to gather and make useful insights from it. Most of the researchers has focused of collecting large amount of test execution data and how to make it readable for non-technical people in different approaches. And this chapter focused on how different authors has done researchers on test automation and different tools they have developed using those methods to cater different problems they have faced using different models.

Chapter 06

Real Time Dashboard for Test Automation in Agile Environment - Objectives

6. Objectives

1. Introduction

This section describes the main objective which will be fulfilled by doing the research as well as all the specific objectives. By developing a customized monitoring real time dashboard for agile test automation development will help lot of agile team members to get useful data from it to make important decisions within the sprint. For product owners, it is important to have a monitoring dashboard to stay focused on the product vision, the needs of the market and stakeholder wants by keeping day to day of quality checks easy to manage. [5] Quick feedback is crucial for agile workflows and making a central dashboard will support all the team members to get quick feedbacks.

2. Main Objectives

To develop a decision making dashboard to help agile team to get useful insights about project and make decisions based test automation results.

1. Specific Objectives

- Update test scripts details real time on dashboard so test automation engineers wont execute all the test scripts on local machine and show the progress or results in their machines to the whole agile development team and also this will help offshore product owners and team to get useful insights of the project progress and can take decisions on product backlog and issues raised.
- Give overview execution of completed test automation scripts.

- Connect with agile management tool and collect test automation execution details for the active sprint.
- Develop a real time dashboard for agile team members to update sprint test automation execution.
- Update sprint automation scripts details on dashboard so that it will save time in daily standups, sprint review meetings, sprint end meetings to track the test automation progress easily.

3. Summary

In this chapter it describes the main objective of this research and the sub objectives trying to achieve from developing the proposed application. As described it is beneficial to the quality engineers as well as to other non-technical people in the agile team and management to make decisions and analyze on the developing product.

Chapter 07

Real Time Dashboard for Test Automation in Agile Environment – Methodology & Implementation

7. Methodology

1. Introduction

This section describes the basic components of the proposed and implemented system, how they are interconnected with each other as well as what technologies will be used to implement the system. And later part of this chapter describes how the system is implemented with the technology stack and tools used to develop this.

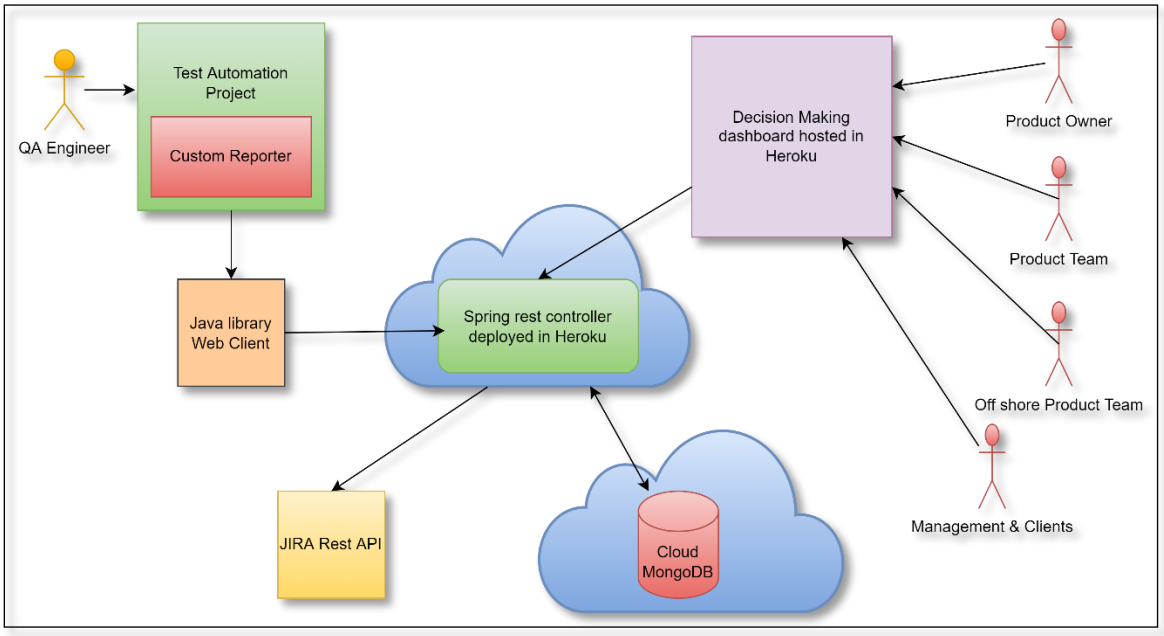

Figure 7.1 High Level diagram of the Overall System

2. Approach

This proposed application is a real time dashboard which updates test automation executions as quality engineers executes and updates it. Along with test execution details real time dashboard will consist of details of active sprint details retrieved from JIRA API to support agile team to use in daily standups and other sprint meetings.

Quality engineers should work in both manual and automation tasks in a sprint for a specific feature and quality engineer has to create a test automation project according to their preference of language and testing framework. In this thesis I have considered web automation project created using testing framework with java and selenium web driver. Additionally I have created a customized reporter to use in test automation project as a base custom reporter. Custom reporter will capture all the execution data from testing listeners and will add it java web client library.

Along with that I have created a custom web client to communicate test automation project and spring boot backend APIs. So we have to add the custom library as a library in automation project. It will help custom reporter to add test execution data to web client and publish it to cloud hosted mongo DB data base.

I have choose cloud hosted mongoDB named mongoDB atlas providing free space in sandbox and can access from anywhere or can configure IP addresses to access it from selected IP addresses. It is easy to manage mongo DB collections as these are NoSQL data bases.

Then to communicate with front end application and test automation project I have created a spring boot back end with REST API. Then created APIs to communicate with JIRA API to get all the active sprint details and publish it on the mongo DB. With these rest APIs system is capable of publishing and retrieving all the test automation execution data to the front end application.

Front end application is a single page application created using angular material and chart.js and bootstrap. It consists of charts tables and mat cards which displays total

number of test cases executed and number of passed test cases, number of failed test cases, and number of skipped test cases in mat cards and shows same data set in a pie chart to give better view. And active Jira sprint details are displayed above the page with active sprint name, sprint start date and sprint end date. In a material table all the test method names and its execution status are published and whenever quality engineer updates same test method it will get updated with test execution status with latest updated time on the table. So then all the product team including quality engineers, product owners, project managers, offshore product team and even the higher management and clients can refer to this real time decision making dashboard to monitor sprint progress and evaluate developing features in a sprint.

As a summary, the application consist of test automation project with a customized java library, test automation API Manager and an operational web dashboard. Test automation engineers will use their IDEs to implement test scripts and need to include customized java library to their projects which will execute and collect test execution results. Then will communicate with API manager to publish test execution results and map it with current sprint details and will display on customized decision making dashboard to get useful data from the test execution results to manage sprint work and will help offshore product owners and higher management to make important decisions.

4. Technologies

To develop proposed decision making dashboard following tools and libraries were used.

Java is used as main programming language in creating test automation project, java web client. TestNG framework used as a test framework with POM model to create basic test automation project and selenium web driver is used to automate. And also used maven as building tool in test automation project.

Then to implement the web client used java and to test web client features used Junit5 Jupiter library to validate. Lombok dependency is used in java web client to reduce boilerplate code for model or data objects. IntelliJ is used as IDE to implement this.

Spring boot REST API is implemented using spring boot to communicate with front end and test automation project and publish data in mongo DB. All the CRUD operations are added here to support communication with each component. This is implemented in ecplise IDE and hosted the spring boot project in Heroku cloud platform which provide services platform as a Service.

I have selected cloud hosted mongo DB so then easy to manage collections and access from anywhere.

Web application is implemented in visual studio code using angular material, chartjs, bootstrap plugins.

As a summary, Rest web client is used to build java library and rest assured library and spring boot to develop API manager, backend is hosted in Heroku platform as a service to access from anywhere, JIRA API Library to map sprint details, Angular material, chart.js, bootstrap and Node.js to develop the dashboard.

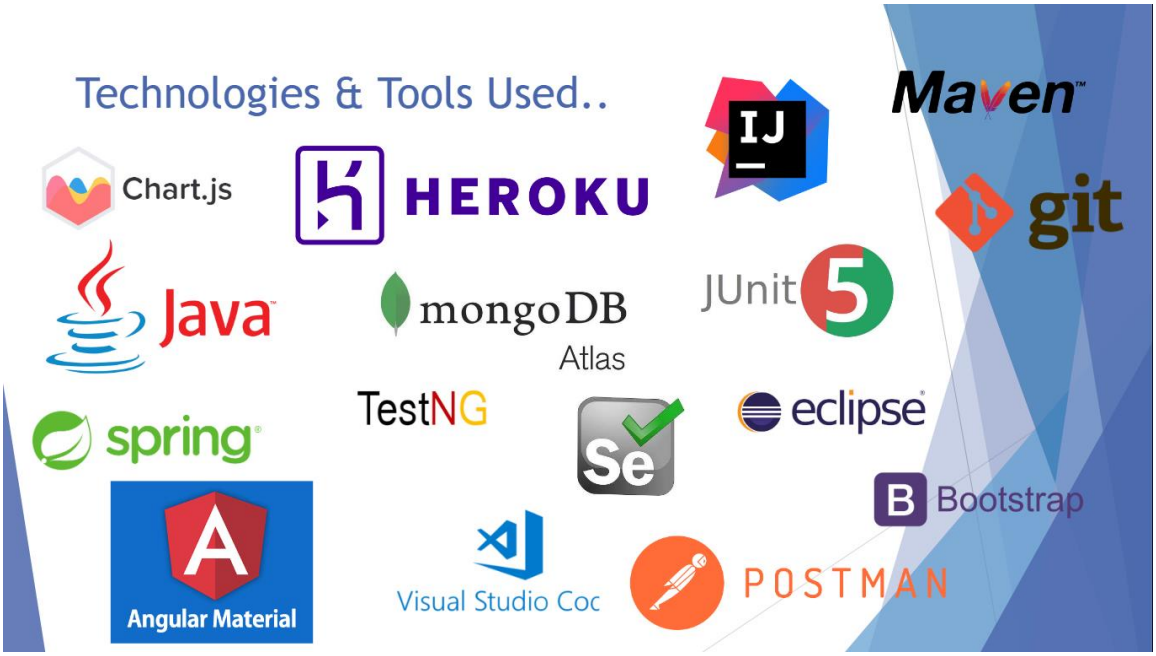

Figure 7.3.1 Used Tools and Technologies

5. Implementation

Quality engineer works on test automation project and whenever they added or updated test cases should be update live on dashboard and should display execution details with a summary report. So to support that I have created customized test automation reporter file to capture automation execution results and relevant data by using different TestNg iListeners available and mapped test methods and execution results in to a Hash maps and defined variables to pass as parameters.

In this proposed project we are intended to capture passed, failed, skipped test methods along with the test execution states and also captures the total number of test methods executed at a time along with number of passed test cases, number of failed test cases and number of skipped test cases.

Moreover, system captures initial test execution time and display in the dashboard and whenever it is executed again system captures updated time and display in the dashboard. These are done using TestNg iListeners and mapping those in to hash maps and customized to cater our requirements.

This customized reporter acts as a kind of base class for execution. Since this is using TestNg listeners, any number of test automation scripts in the automation project could execute and update the dashboard as we are using '@Test' annotation from TestNG in each executable tests in our test scripts and we are using TestNG listeners to capture execution details. There for all the executable test methods will be captured by custom reporter and will update the real time decision making dashboard every time quality engineer updates.

Along with the custom reporter, Quality Assurance engineer has to import and add the customized java web client library developed to the test automation project in order to communicate with Rest API backend hosted in heroku platform. Java library acts as a rest web client which synchronously communicates with rest API, but have used only one way communication for the above requirements. To develop this I have used Rest Web client methodologies, java as main developing language and to test rest web client API requests I have used Junit 5 Jupiter library and Lombok functions. Used Junit Jupiter to

test all these HTTP requests from java library. Java library is developed using java, web client with Data Transfer Objects. Used InteliJ IDE to develop this rest web client java library.

Test automation project is developed using java, selenium web driver and TestNg framework in IntelliJ IDE. And consists with sample test methods with assertions to demonstrate the passed, failed, skipped scenarios.

Then created Java library as a Rest web client to communicate with rest API hosted in heroku cloud platform and with the test automation project. With the java rest web client, intended test results will be passed to the Rest API backend hosted in heroku platform and then will publish execution data in cloud mongo db collections.

As a database to the proposed system, used cloud hosted mongodb database "MongoDB Atlas". All the data passed from java web client will be publish in the mongodb using rest APIs in different collections.

Rest API is developed to communicate with java web client and cloud MongoDB. Rest API is deployed in Heroku in order to make rest http requests so then it will be easier with all the automation engineers to update all their test scripts via one hosted place. Rest API is developed using Spring Web, Rest controllers with connected to cloud mongoDB atlas

To develop the frontend I have used angular material, chart.js and bootstrap and deployed in heroku to make it available in a specific domain to access by everyone. I have used Visual studio IDE to develop this frontend application. Initially all the http requests tested in postman in local host and local mongodb. Then all backend API application and frontend application are hosted in Heroku application.

To use this application, Quality Assurance engineer/ Automation Test Engineer has to add this custom test reporter in their test automation project and add the JAR in to their project path. Then when Quality Assurance engineer/ Automation Test Engineer execute their test automation scripts it will automatically get update in the real time Dashboard. So then other non-technical people and agile product team members could refer to this web page and get better insights about developing application.

6. Testing the implementation

Unit testing is a level of software where individual components of software are tested. The purpose is to validate that each software component is working according to the user requirement.

In testing phase, input data validation, whether the selected data is stored in the database, generated data is accurate are tested. In the implemented system all the testing components were tested in component level. And then tested system as a whole to validate the system is functioning properly.

Initially the custom report were tested with sample data whether we get the correct data from test execution. All the sample data were logged and tested each functionality alone.

And the tested the java library with web client with Junit 5 testing framework to test the web client synchronous methods working fine. For that especially I used Junit 5 Jupiter library to test as it is the only library to test upgraded web client http requests. For all the web client functions unit tests have been written and all the unit tests are passed.

Furthermore, tested the java library with custom test reporter. Sprint rest controller with all the CRUD operations first tested in local machine with local mongoDB using postman application. Then after deployed in the heroku tested in cloud mongo DB atlas application. The angular application is tested initially in the local host and then deployed to heroku.

7. Summary

This chapter comprehensively describes the way of implemented of this proposed application and the technologies and tech stack used to implement it successfully. And also it discuss how to use it and pre configurations which needs to be done before using this application in test automation projects in agile environment.

Chapter 08

Real Time Dashboard for Test Automation in Agile Environment – Future research Work

8. Future work & Limitations

This application is just an initial approach to implement real time decision making dashboard for test automation development focusing on Software Test automation development progress in an agile team. This application could be improved more on adding more test execution results to the dashboard with features like total execution time and exceptions in the failures to get an idea about the failures. And also to track the already developed scripts and execution history by storing results in AWS S3 buckets in each time frame which enables to download as reports from S3 bucket. So then can add another feature to download reports for past sprints in the agile project.

This application is developed considering a limited scope; some parts are considered as limitations of this project. There is a limitation to find whether actually test scenarios is completed. Some may develop a half of the test script to be successfully pass but yet it is not completed the whole scenario written on the manual test case. That will miscommunicate with team as a successful test script in the dashboard. In future enthusiast researcher could work on this area to overcome this limitation.

Moreover, future researchers can add some features to save real time updating test scripts to save in the cloud and execute those test scripts hourly in a grid to generate and get failure patterns and loading issues in peak and off peak hours and display in the same dashboard to make this more usable and interactive with agile team members.

Chapter 09

Real Time Dashboard for Test Automation in Agile Environment - Conclusion

9. Conclusion

The outcome of this research is developing a decision making dashboard for agile Test Automation Development to identify the sprint test automation details and get useful insights of it. This dashboard will act as a supportive monitoring dashboard for all the members in the agile development team as well as the product owners and management in offshore. Moreover, this application helps test automation engineers to provide update while they working on sprint items without executing repeatedly in local machine. Automation engineers test scripts will update the dashboard in real time so all the agile team members can check the dashboard while automation engineers are working on automation scripts. And also this application will help product owners and scrum masters how development team is performing during the sprint and the failures identified, defects identified and deployment issues in the developing feature and if it's getting more delays product owner can refine product backlog items and prioritize accordingly to plan future sprint work. However, author shows the importance of re using test automation results and getting benefits of them such as reusability, repeatability and effort saved in test execution and to propose an orderly new way of thinking and organizing all test automation artifacts. [15] Finally, it a pleasure to implement innovative application using test automation artifacts to produce novel product and gain more valuable outcome from it.

References

[1] M. Mulders, “Test Automation Benefits: 12 Reasons to Automate in 2020.” https://www.testim.io/blog/test-automation-benefits/ (accessed Jul. 09, 2021).

[2] H. Upreti, “A Dashboard is Worth a Thousand Tests.” https://smartbear.com/de/blog/2017/a-dashboard-is-worth-a-thousand-tests/ (accessed Jul. 09, 2021).

[3] Loke Mun Sei, “Automating Test Activities: Test Cases Creation, Test Execution, and Test Reporting with Multiple Test Automation Tools,” 2015. https://publications.waset.org/10002771/automating-test-activities-test-cases-creation-test-execution-and-test-reporting-with-multiple-test-automation-tools (accessed Jul. 12, 2021).

[4] M. F. Dorothy Graham, Ed., Experiences of Test Automation: Case Studies of Software Test Automation - Dorothy Graham, Mark Fewster - Google Books.

[5] “Test Results Dashboard for Product Owners | Calliope Pro.” https://www.calliope.pro/for-product-owners/ (accessed Jul. 11, 2021).

[6] “Pricing | Calliope Pro.” https://www.calliope.pro/pricing/ (accessed Jul. 11, 2021).

[7] “Features | Calliope Pro.” https://www.calliope.pro/features/ (accessed Jul. 11, 2021).

[8] “Test Reporting & Analytics | Automation Dashboard | Perfecto.” https://www.perfecto.io/platform/analytics-reporting (accessed Jul. 12, 2021).

[9] “Test Reporting & Analytics | Automation Dashboard | Perfecto.” https://www.perfecto.io/platform/analytics-reporting (accessed Jul. 13, 2021).

[10] “Test Automation Data for Agile Environments.” https://www.infocepts.com/blog/test-automation-for-agile-environments/ (accessed Jul. 16, 2021).

[11] X. Zhu, B. Zhou, J. Li and Q. Gao, "A test automation solution on GUI functional test," 2008 6th IEEE International Conference on Industrial Informatics, 2008, pp. 1413-1418, doi: 10.1109/INDIN.2008.4618325.

[12] H. Miquel Vidal Dura, “VISUALIZATION OF SOFTWARE DEVELOPMENT METRICS ON A DASHBOARD.”

[13] V. Russo, A. Oriou, F. Huynh, and C. Baron, “Software Quality Assurance Dashboard for Renault Software Robustness plan with SQUORE tool Software Quality Assurance Dash-board for Renault Software Robustness plan with SQUORE tool. 9th European Congress EMBED-DED REAL TIME SOFTWARE AND SYSTEMS (ERTS Software Quality Assurance Dashboard for Renault Software Robustness plan with SQUORE tool,” 2018, Accessed: Jun. 04, 2021. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01708535v2.

[14] Y. Wang and M. V Mäntylä, “Test Automation Process Improvement in a DevOps Team: Experience Report,” 2020.

[15] D. Almog, H. S. Chassidim, Y. Tsubery, M. Bures, and S. Mark, “Conceptual Approach for Reuse of Test Automation Artifacts on Various Architectural Levels,” doi: 10.2298/CSIS170701006A.

[16] S. Ibarra and M. Muñoz, "Support tool for software quality assurance in software development," 2018 7th International Conference On Software Process Improvement (CIMPS), 2018, pp. 13-19, doi: 10.1109/CIMPS.2018.8625617.

[17] M. Staron, W. Meding, J. Hansson, C. Höglund, K. Niesel, and V. Bergmann, “Dashboards for continuous monitoring of quality for software product under development.”

[18] J. W. Yoder and R. Wirfs-Brock, “QA to AQ Part Two Shifting from Quality Assurance to Agile Quality ‘Measuring and Monitoring Quality,’” 2014.

[19] A. Agarwal, N. K. Garg and A. Jain, "Quality assurance for Product development using Agile," 2014 International Conference on Reliability Optimization and

Information Technology (ICROIT), 2014, pp. 44-47, doi: 10.1109/ICROIT.2014.6798281.

[20] P. Kumar, A. Kumar Rathaur, R. Ahmad, M. K. Sinha and R. Sangal, "Dashboard: An integration and testing platform based on backboard architecture for NLP applications," Proceedings of the 6th International Conference on Natural Language Processing and Knowledge Engineering(NLPKE-2010), 2010, pp. 1-8, doi: 10.1109/NLPKE.2010.5587779.

[21] H. Demirkan, "Towards a Theory of Agile Dashboards for Service Oriented Organizations," Accessed: Jun. 20, 2022. [Online]. Available: http://aisel.aisnet.org/amcis2009/683.

[22] L. López *et al.*, "QaSD: A Quality-aware Strategic Dashboard for supporting Decision makers in Agile Software Development," doi: 10.1016/j.scico.2020.102568.

Appendix

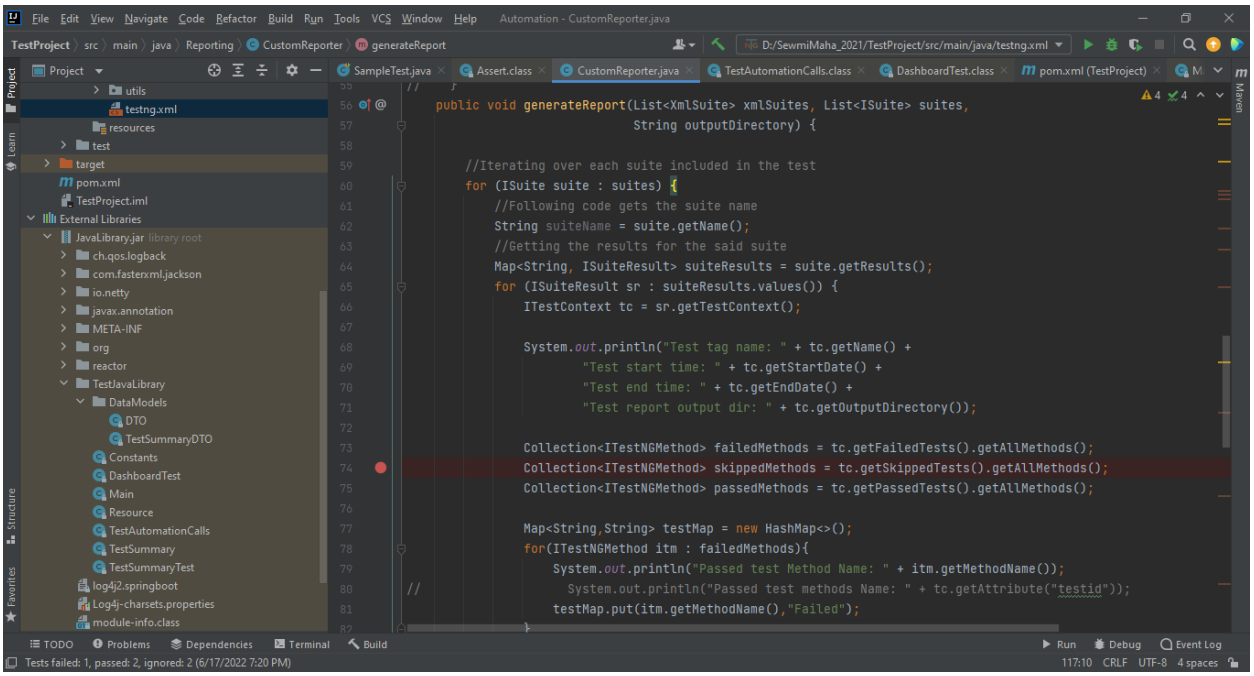

Figure 7.4.1 Customized test reporter

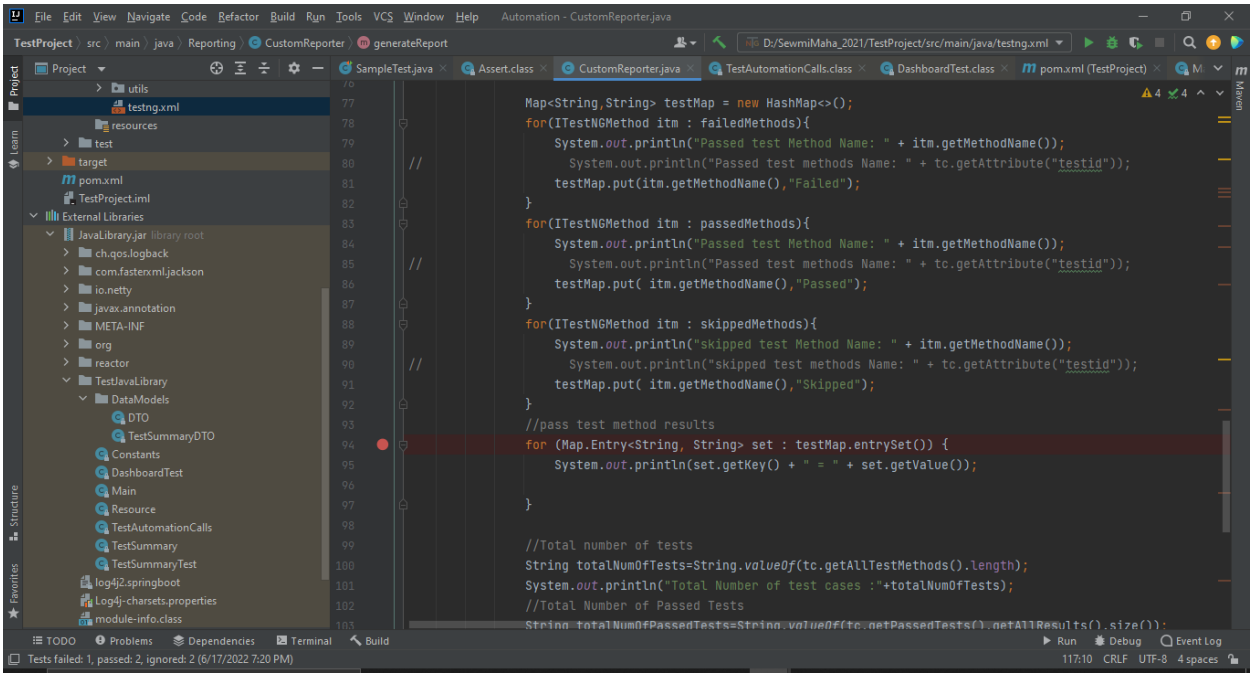

Figure 7.4.2 Customized test reporter other test data

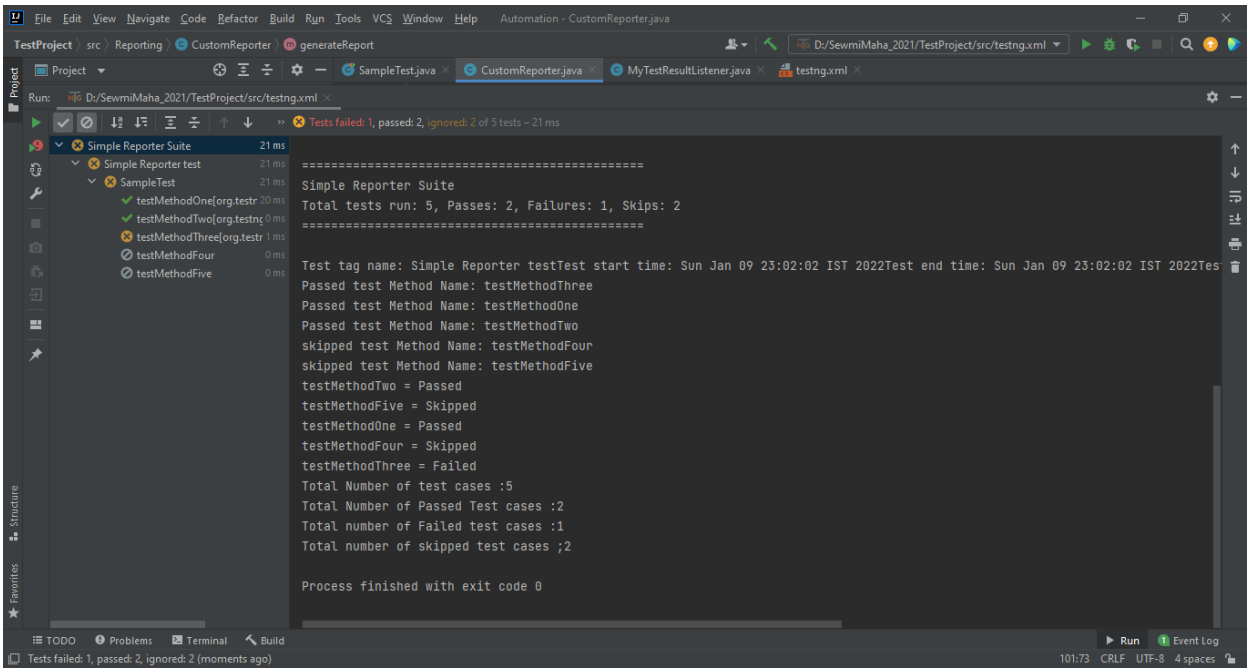

Figure 7.4.3 Customized test reporter test results captured in console

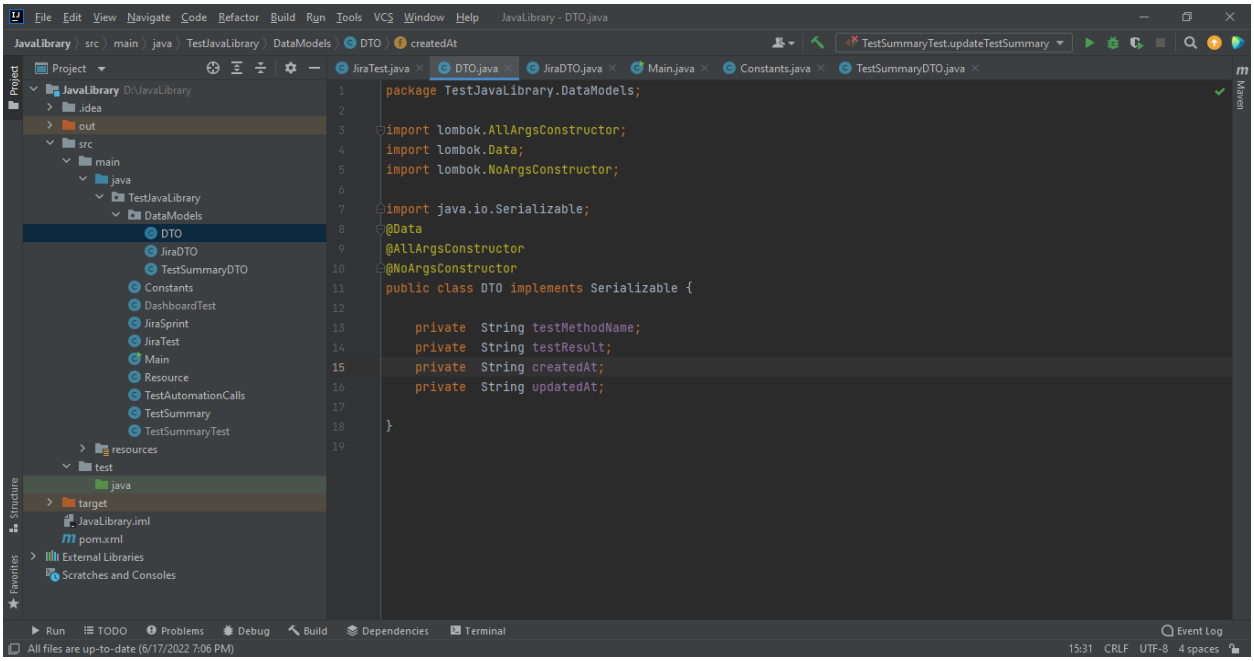

Figure 7.4.4 Web Client java library to post data to API manager

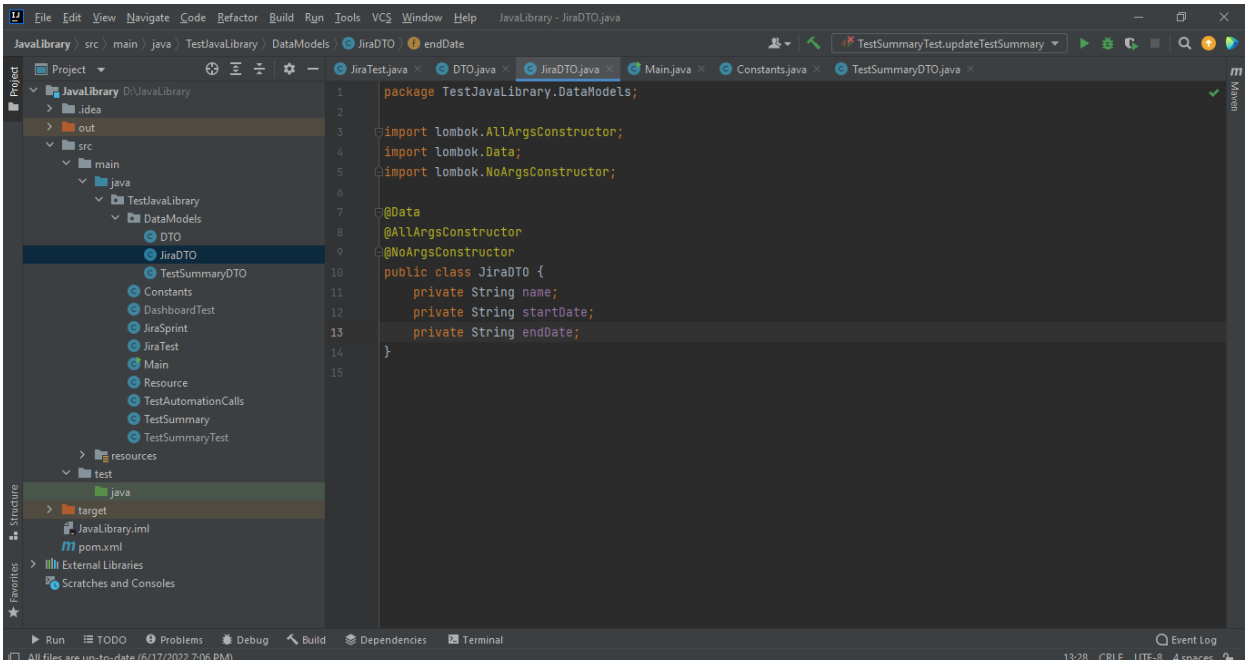

Figure 7.4.5 Web Client java library to post JIRA data to API manager

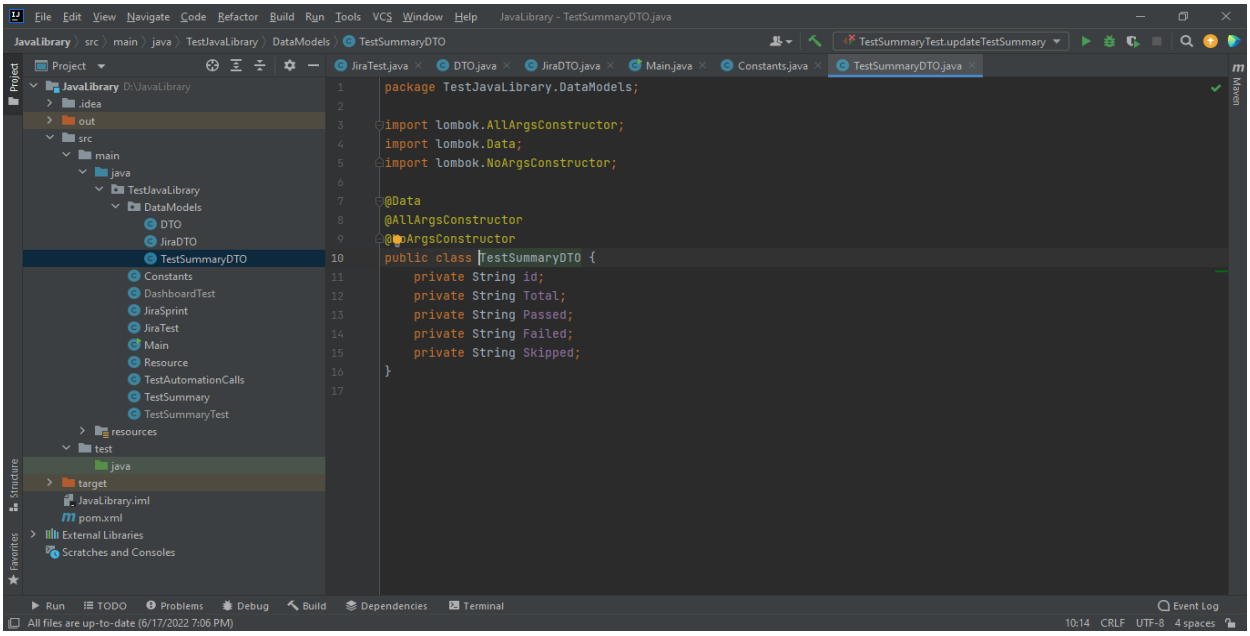

Figure 7.4.6 Web Client java library to post Test Summary data to API manager

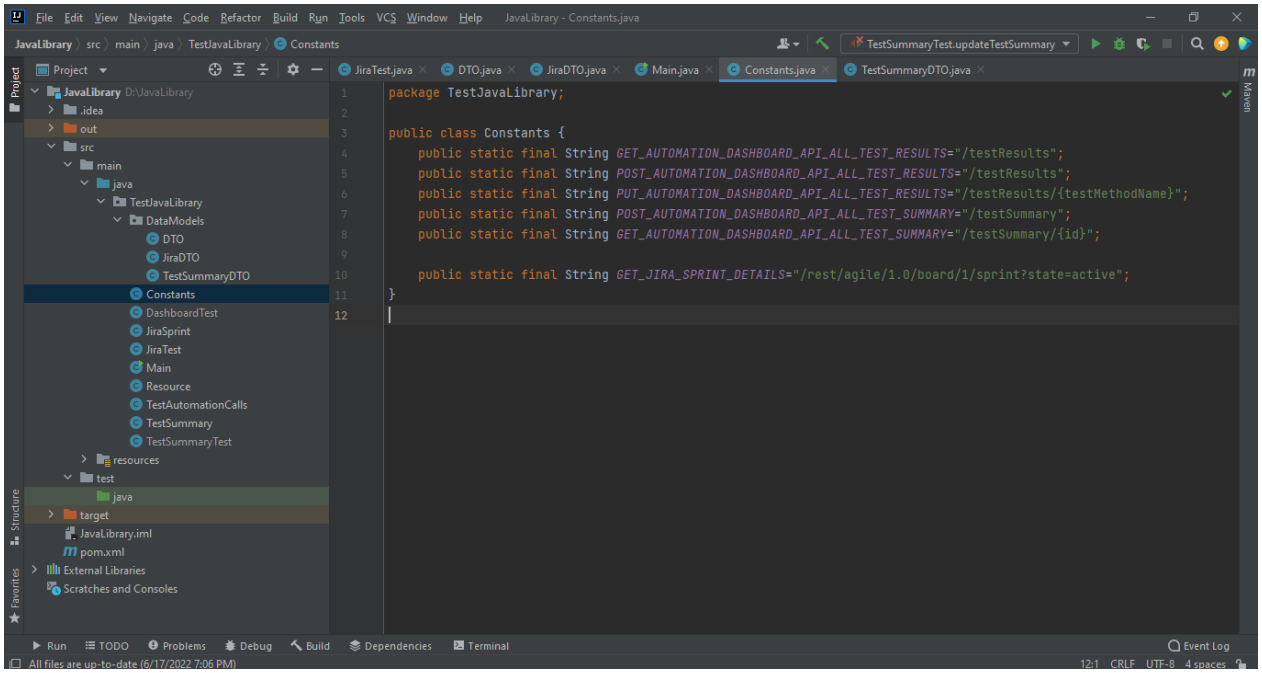

Figure 7.4.7 API requests in Constant file

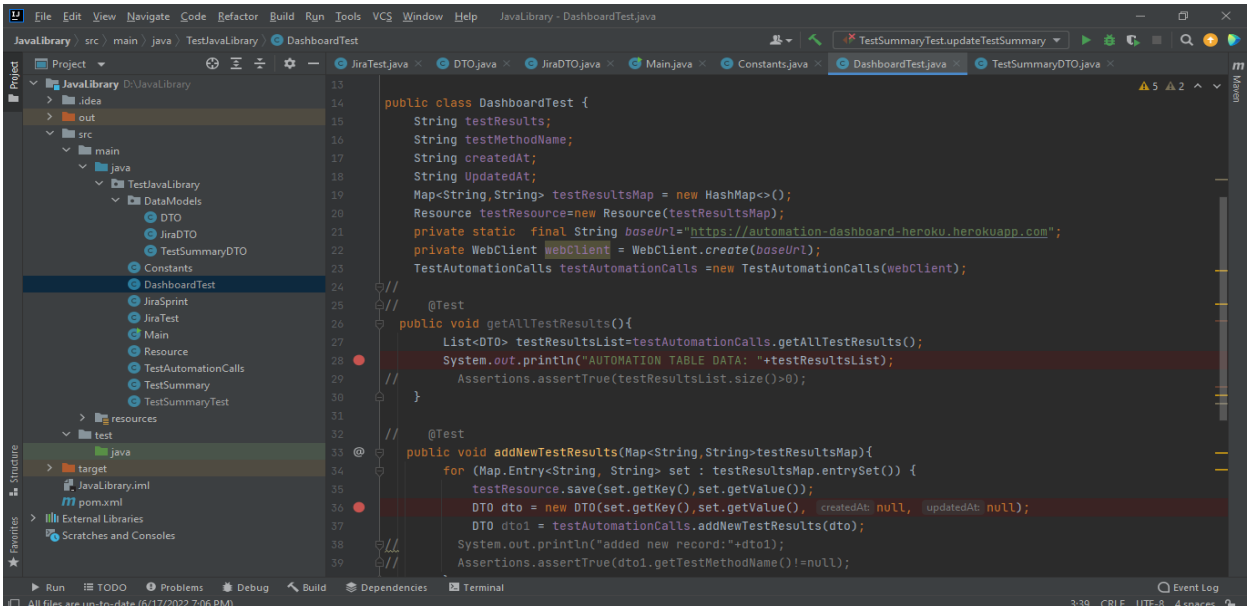

Figure 7.4.8 Web Client to get test script data from automation project

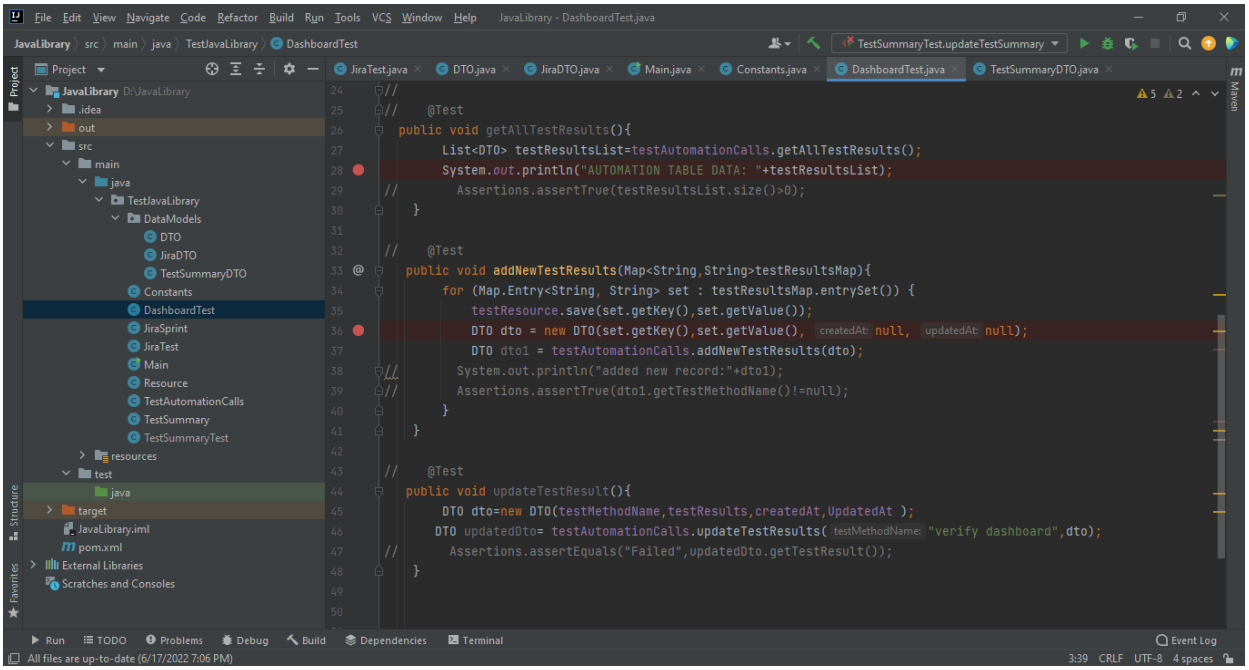

Figure 7.4.9 Web Client to get test script data from automation project_02

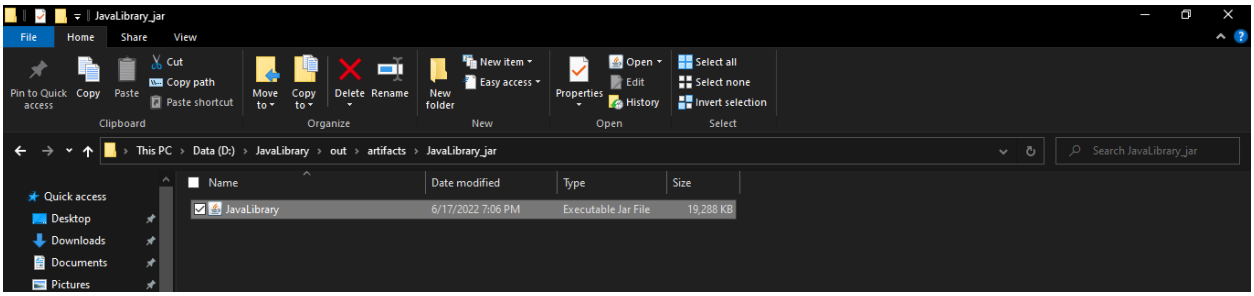

Figure 7.4.10 Completed JAR file using Maven

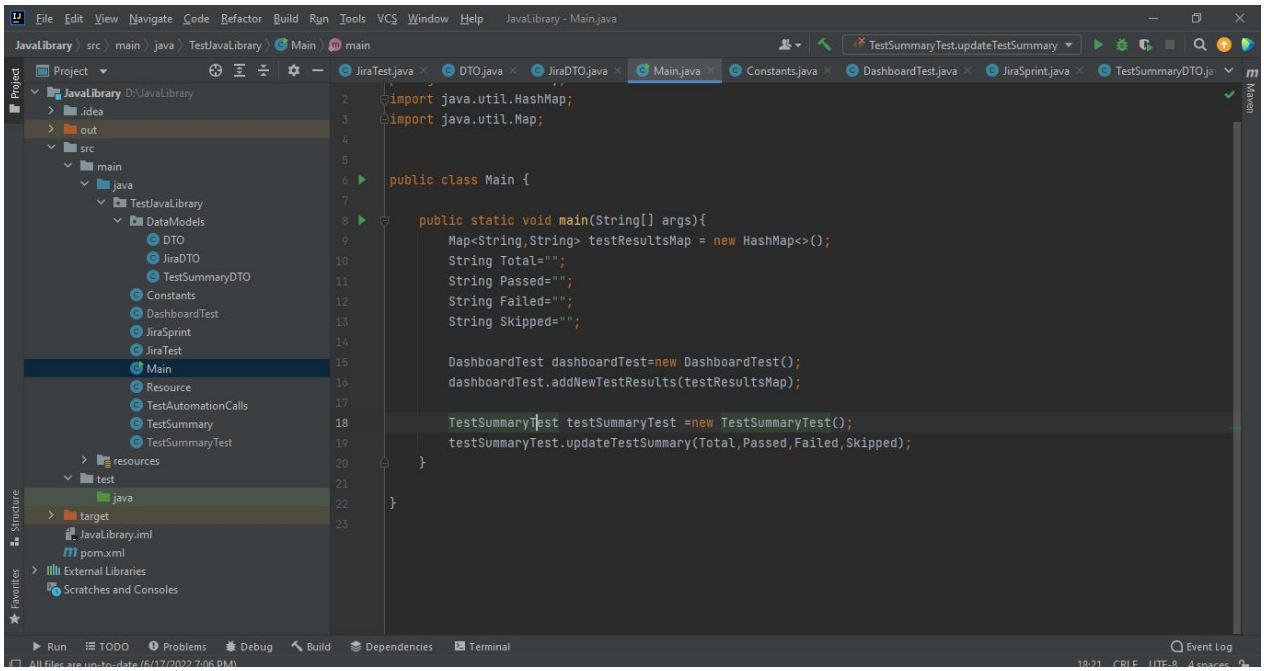

Figure 7.4.11 Main method

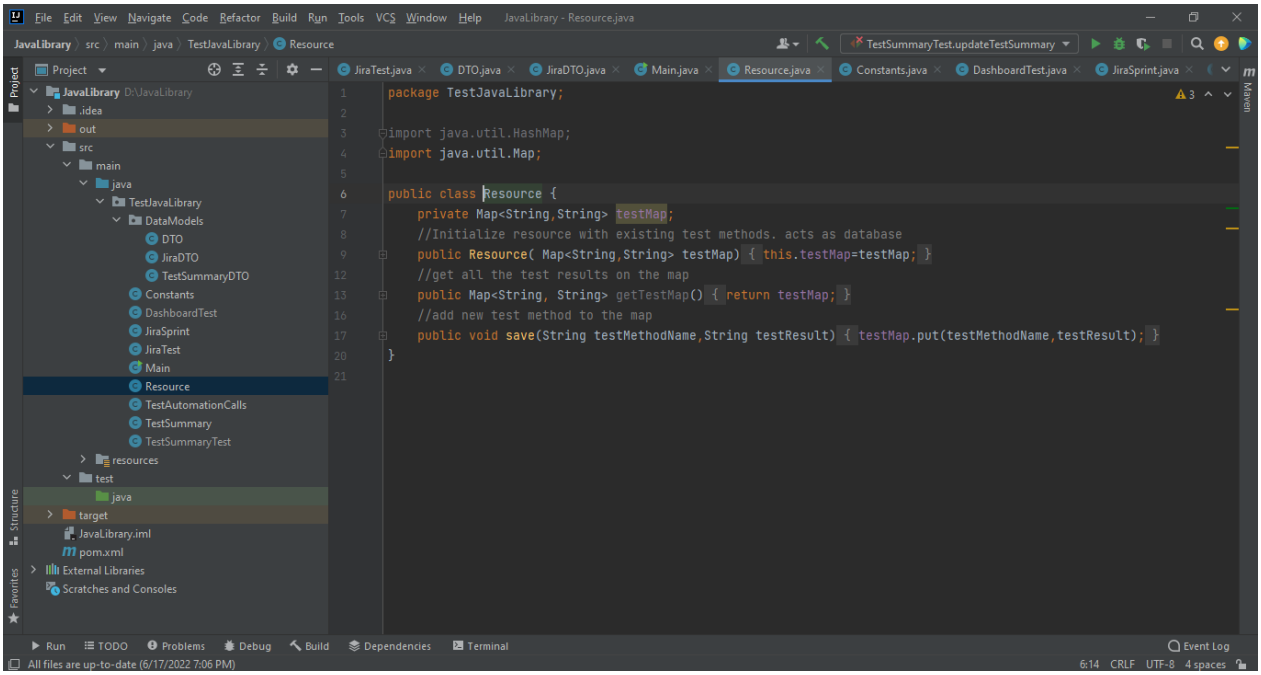

Figure 7.4.12 Resources

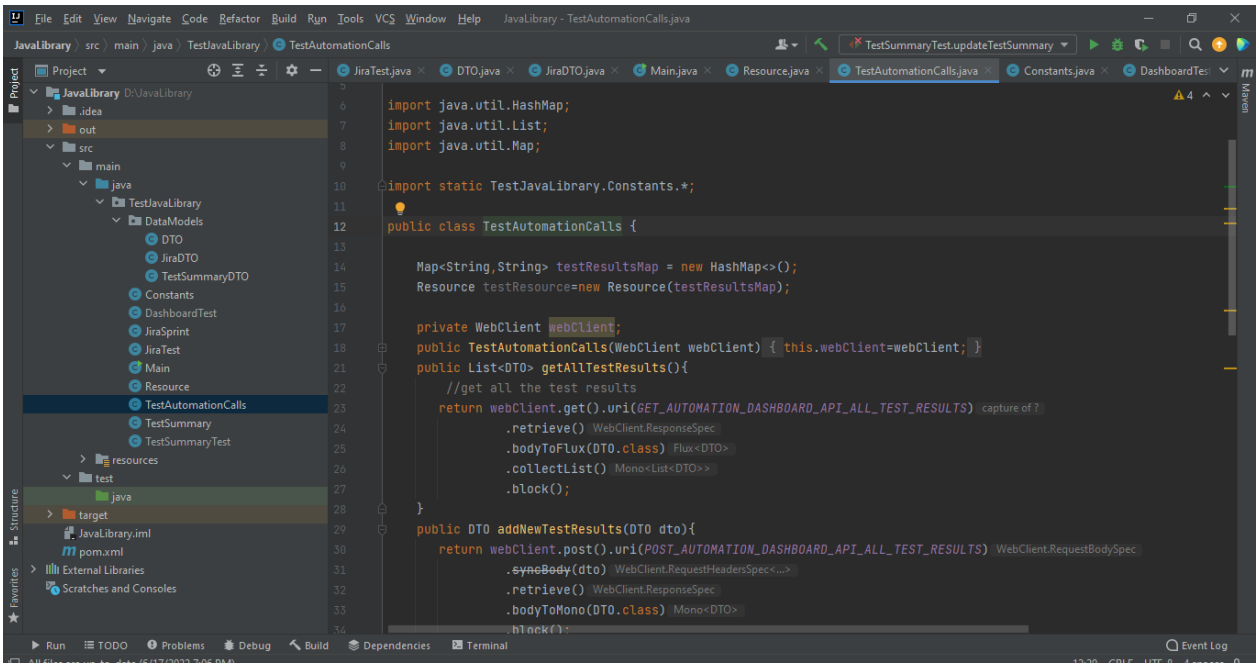

Figure 7.4.13 Web client to publish test automation data to Rest API

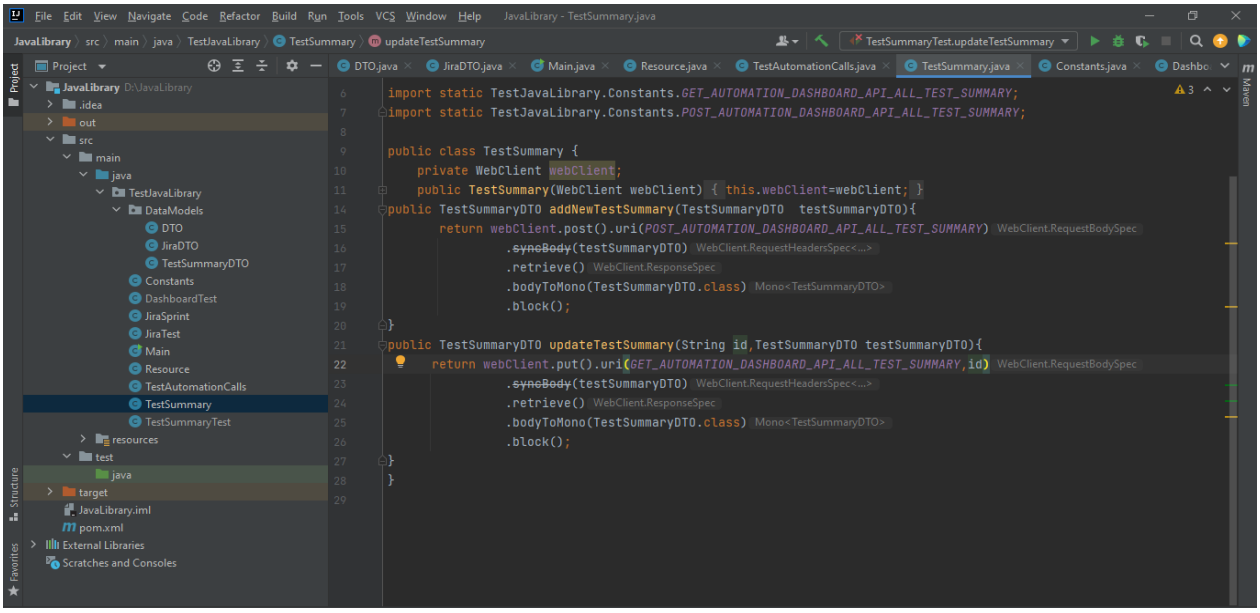

Figure 7.4.14 Web client to publish test Summary data to Rest API

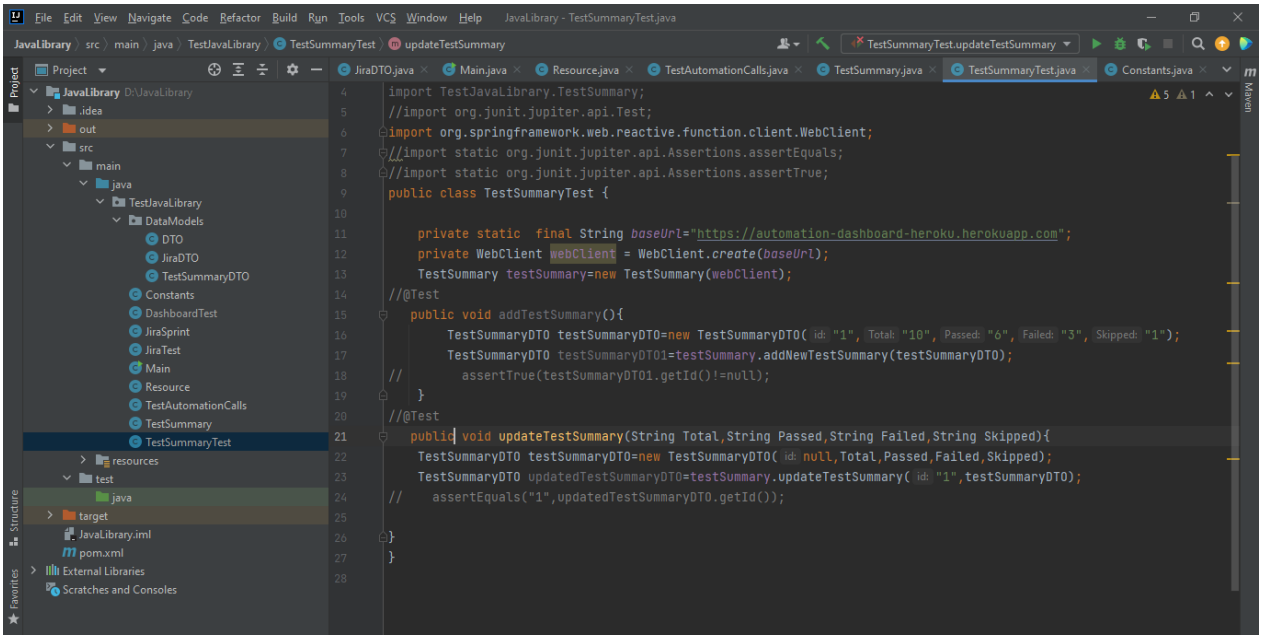

Figure 7.4.15 Publish test summary to Rest API

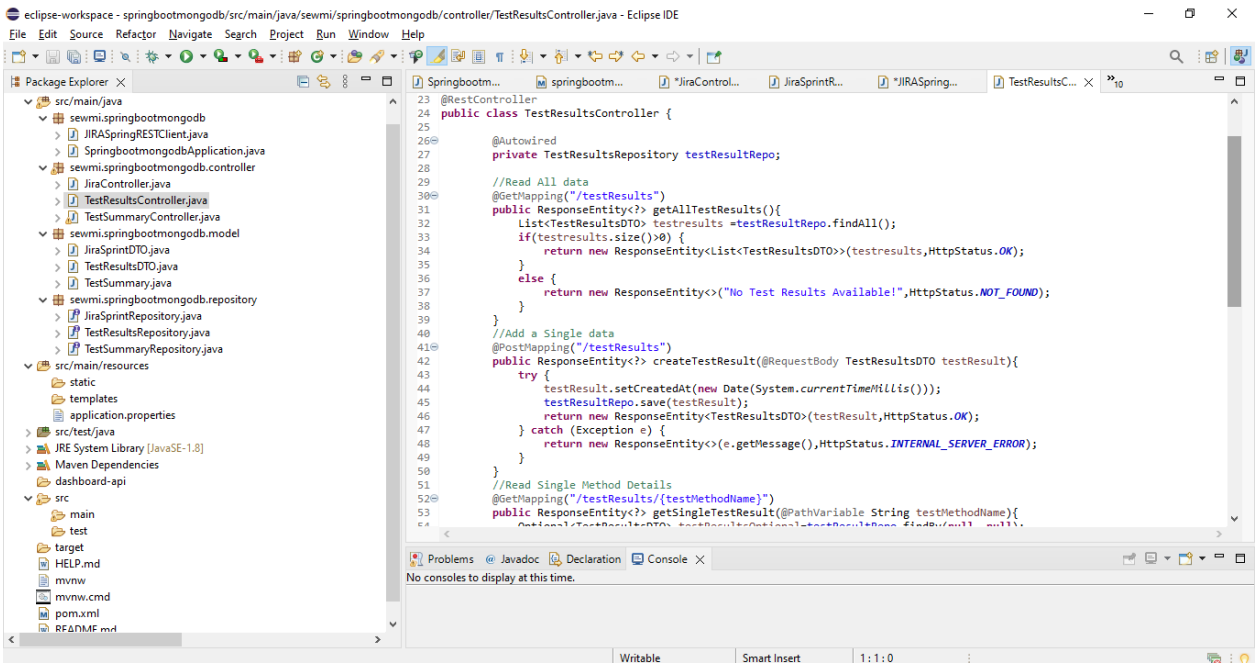

Figure 7.4.16 Spring boot – Rest controller for test results

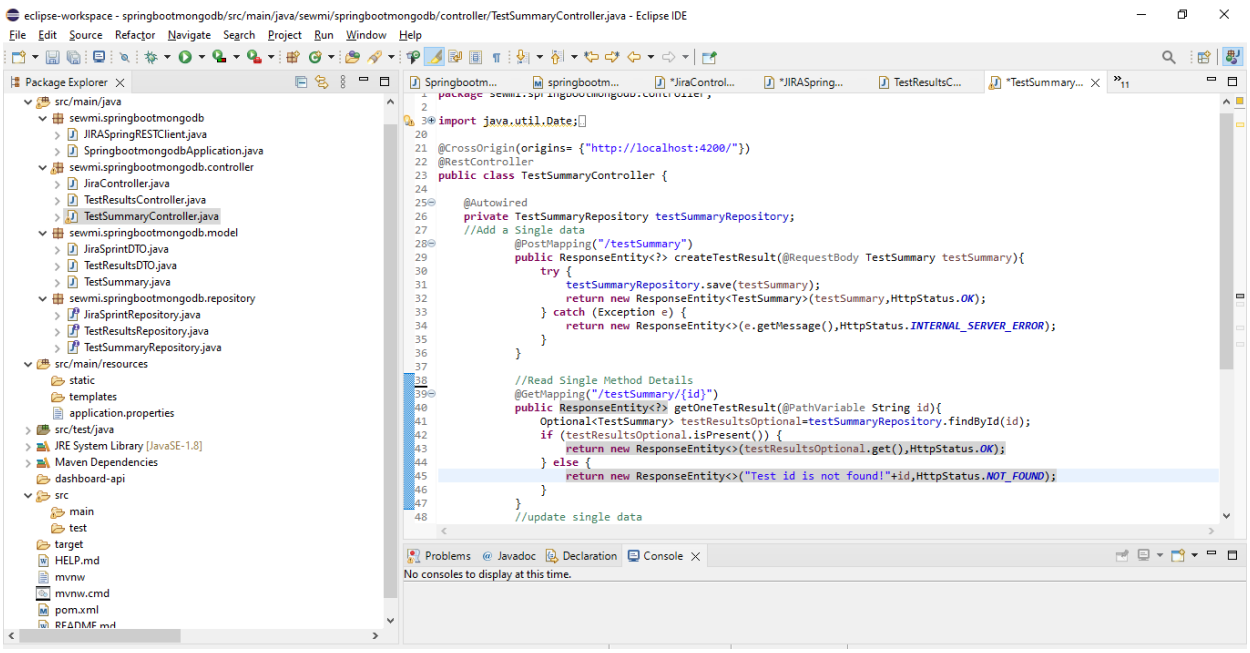

Figure 7.4.17 Spring boot – Rest controller for test results

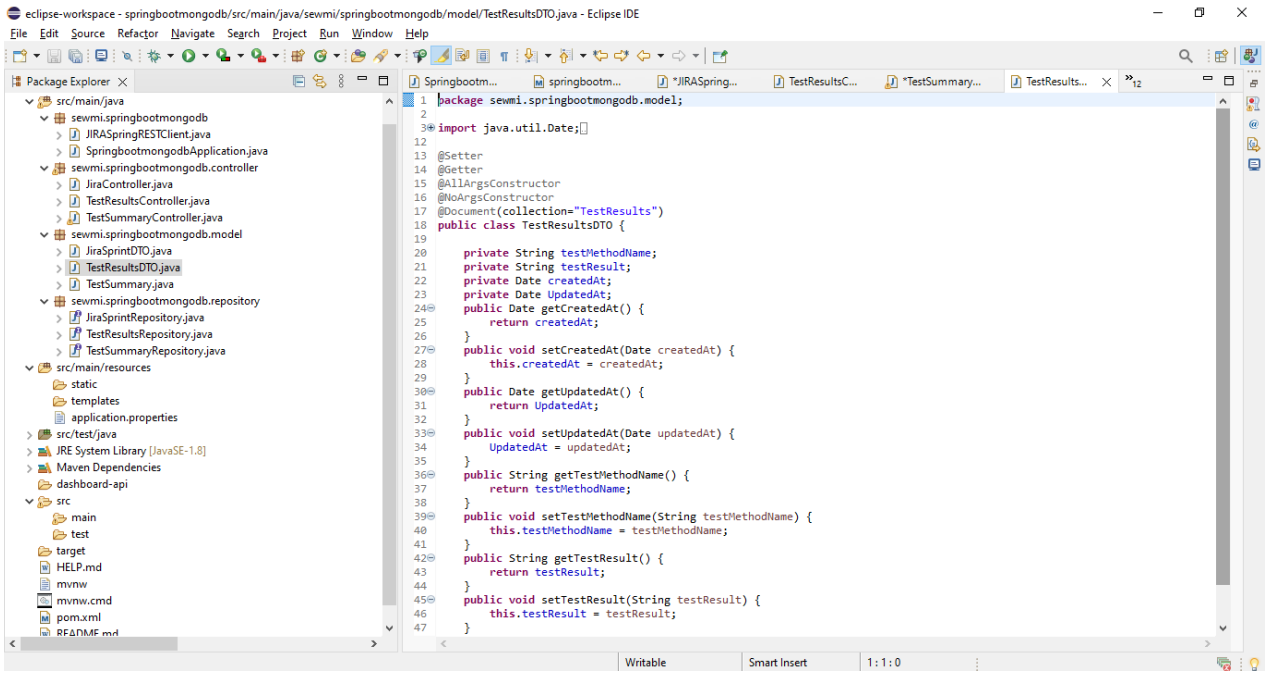

Figure 7.4.18 Spring boot – DTO for test results

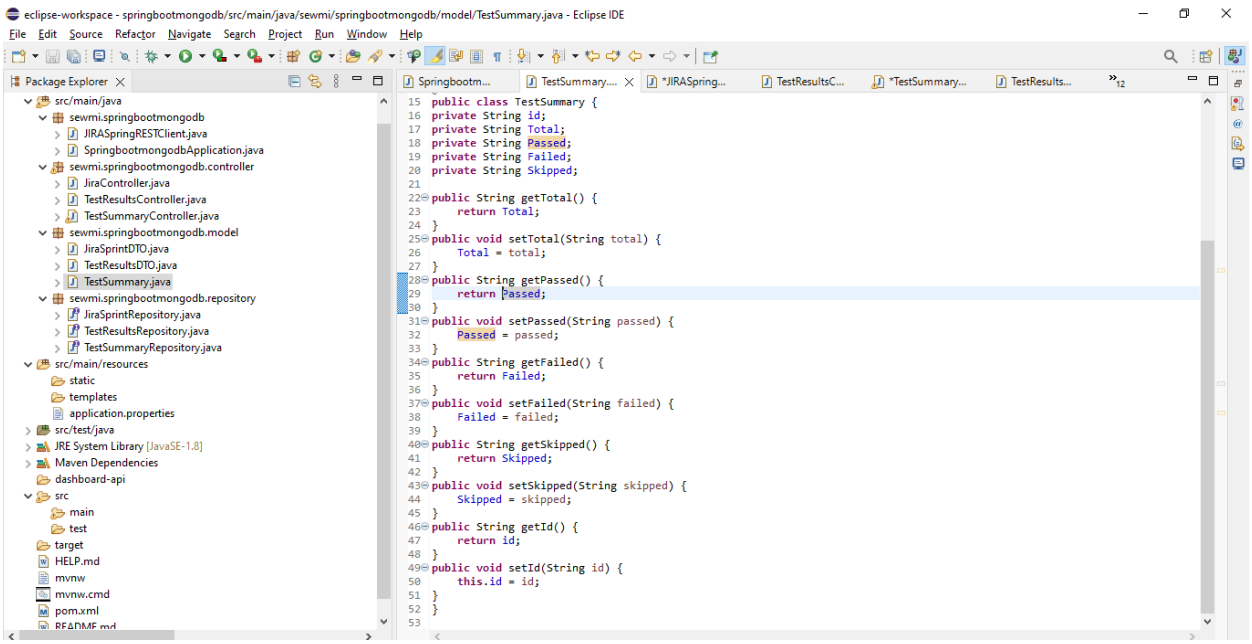

Figure 7.4.19 Spring boot – DTO for test summary

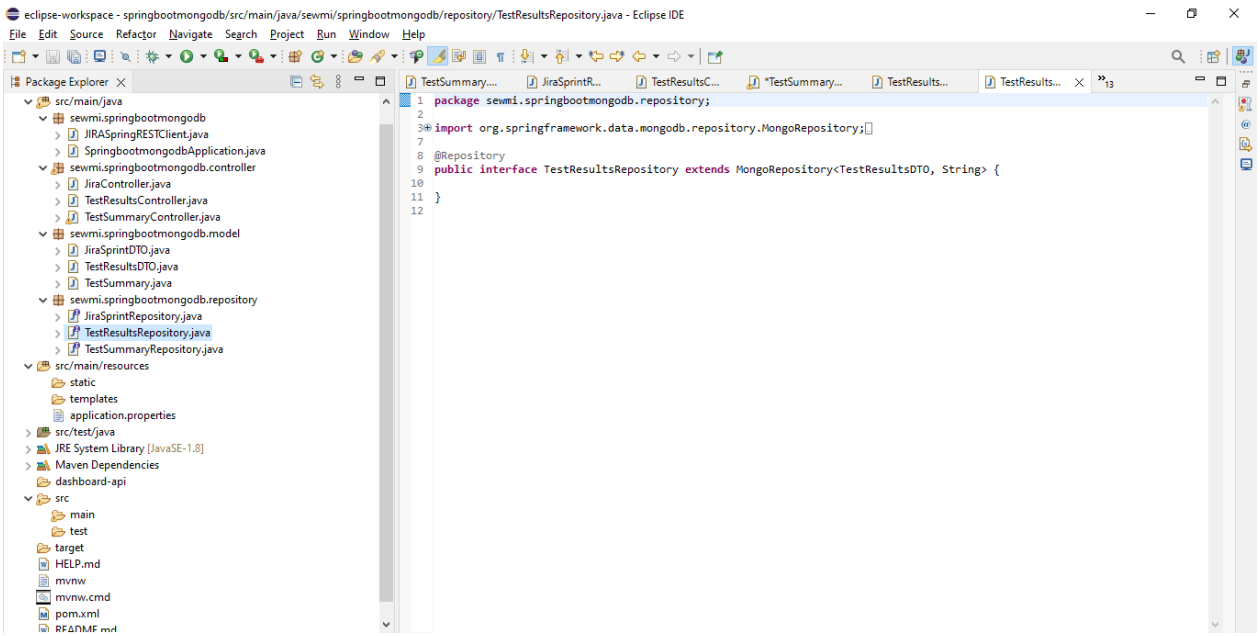

Figure 7.4.20 Spring boot – Repository for test results

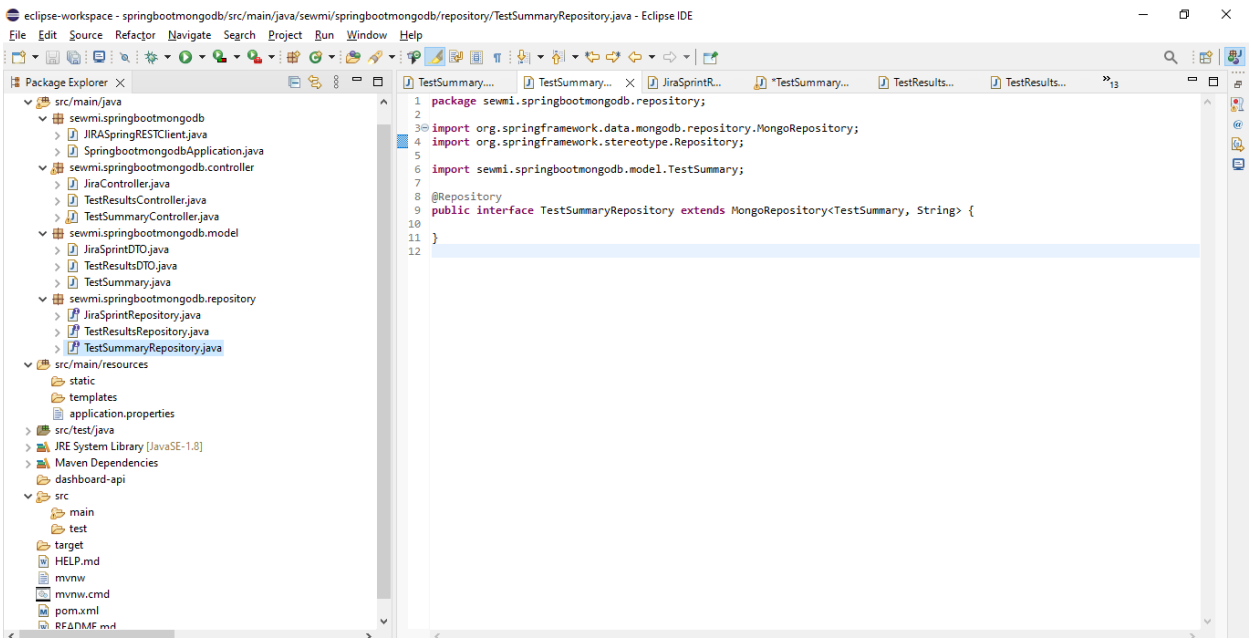

Figure 7.4.21 Spring boot – Repository for test summary

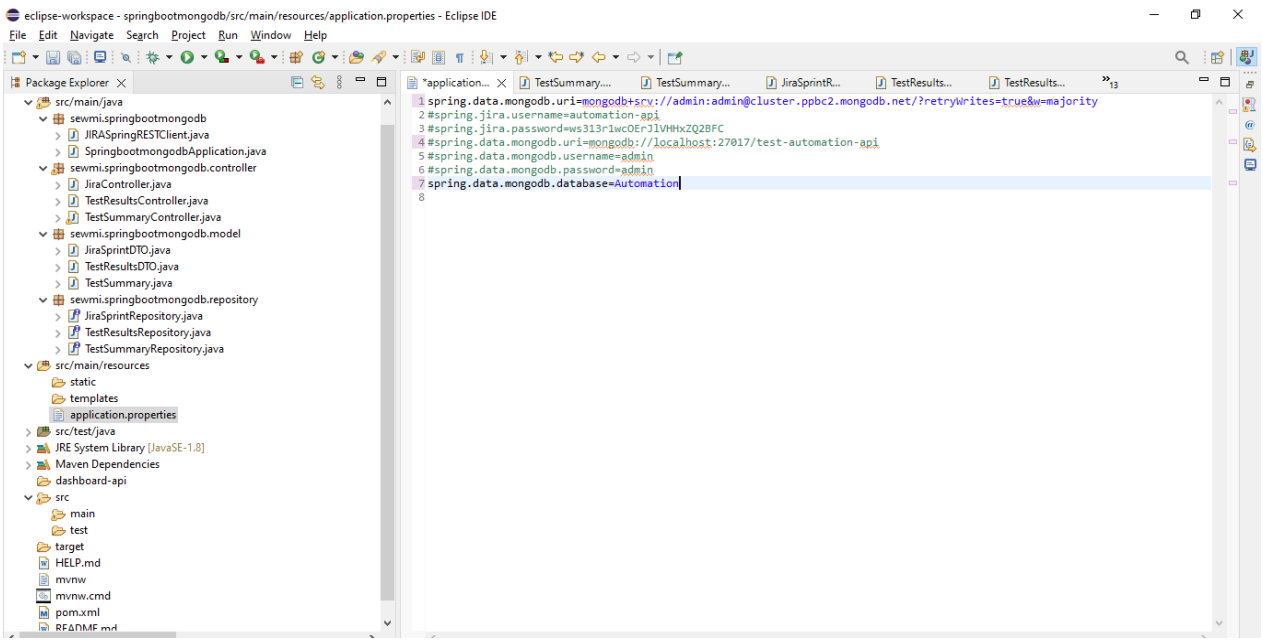

Figure 7.4.22 Spring boot – application properties

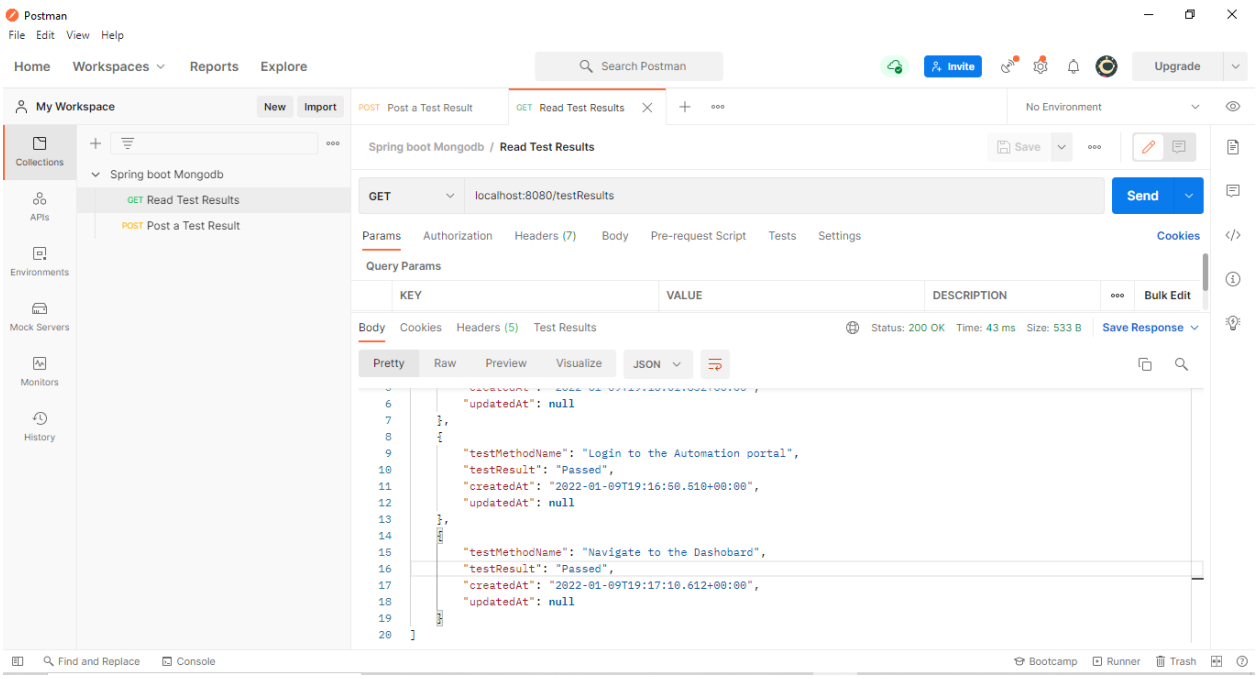

Figure 7.4.23 Postman to check API responses in local host- Initial testing

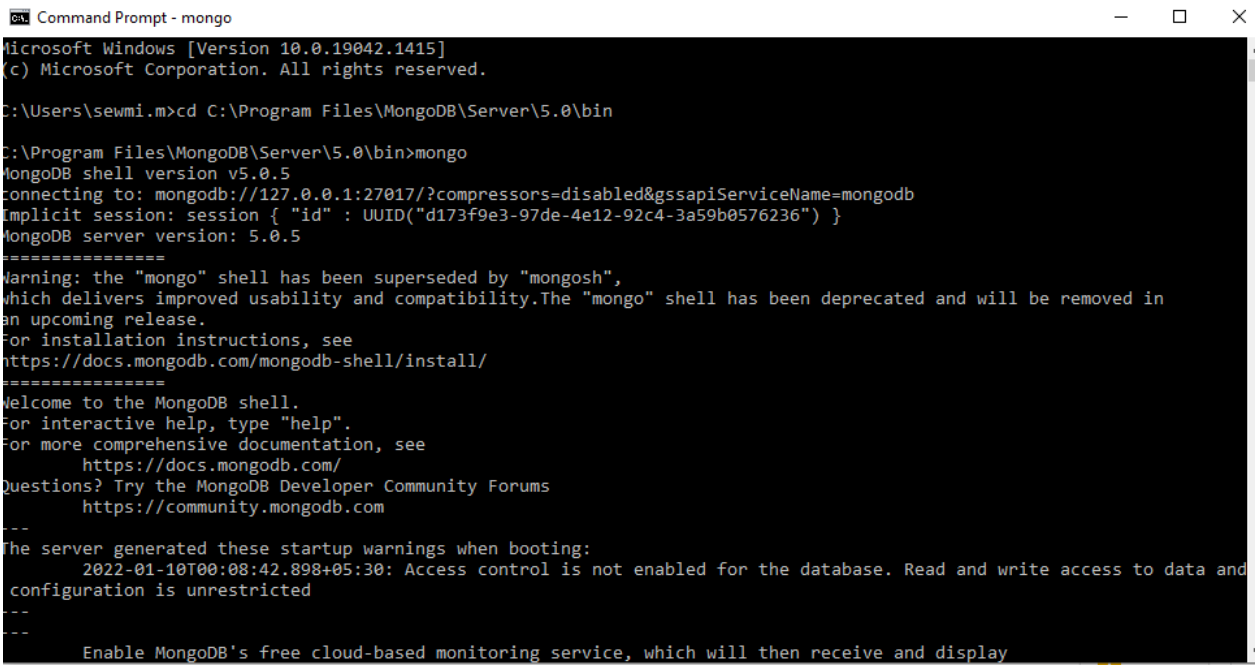

Figure 7.4.24 Connected to mongo db server to connect API

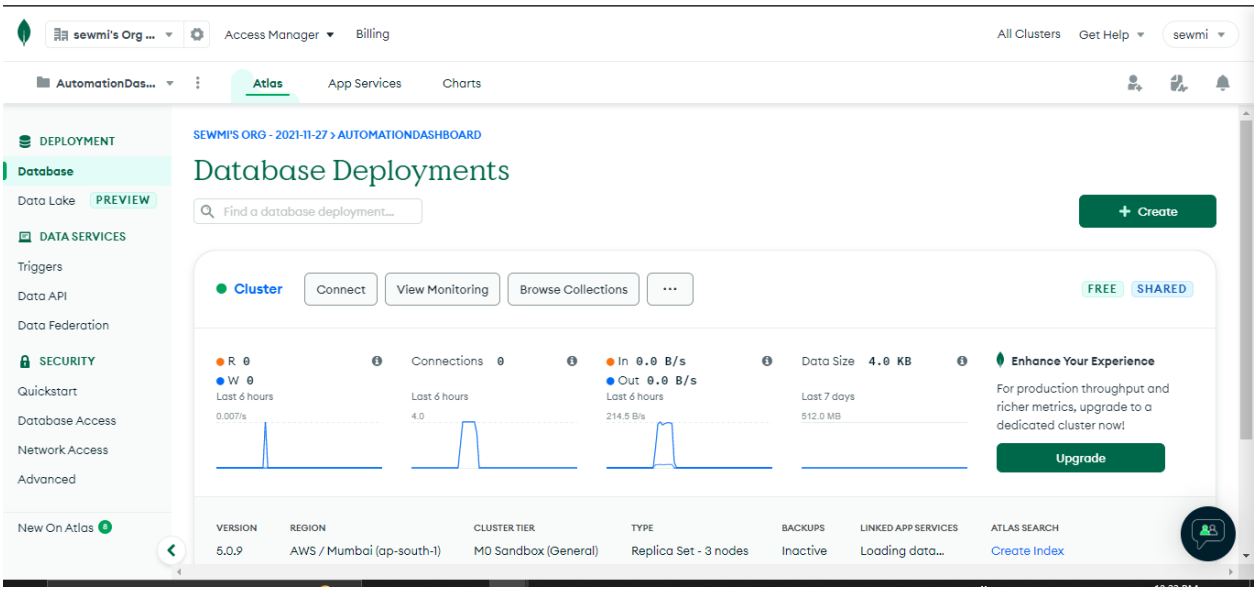

Figure 7.4.25 Connected to Cloud mongo db server

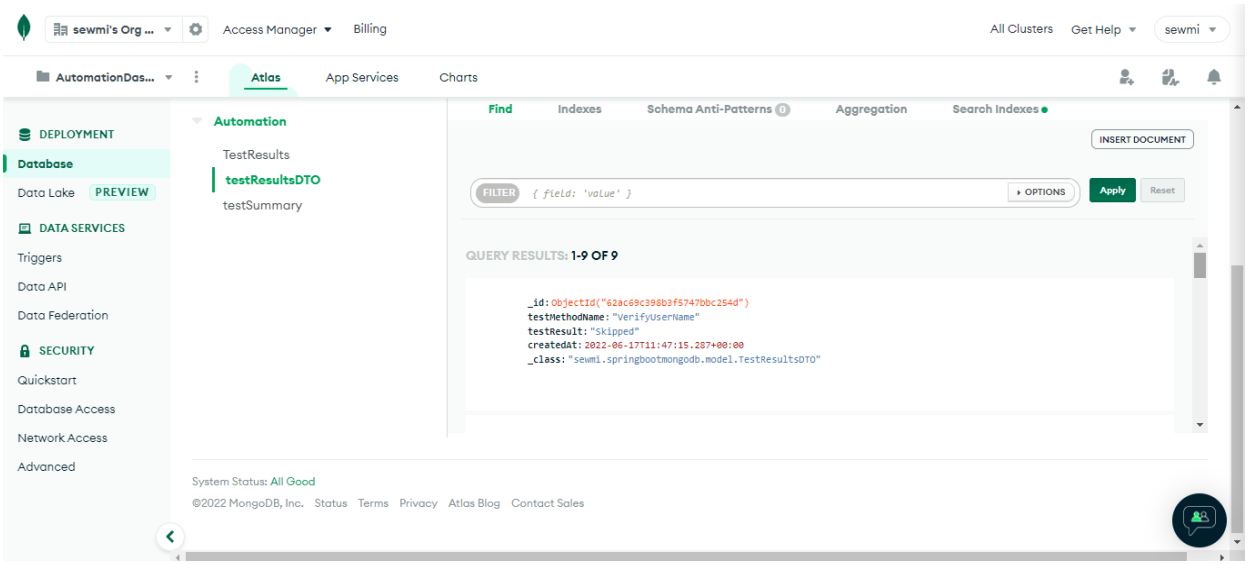

Figure 7.4.26 Data Collections – test automation data in cloud mongo db atlas

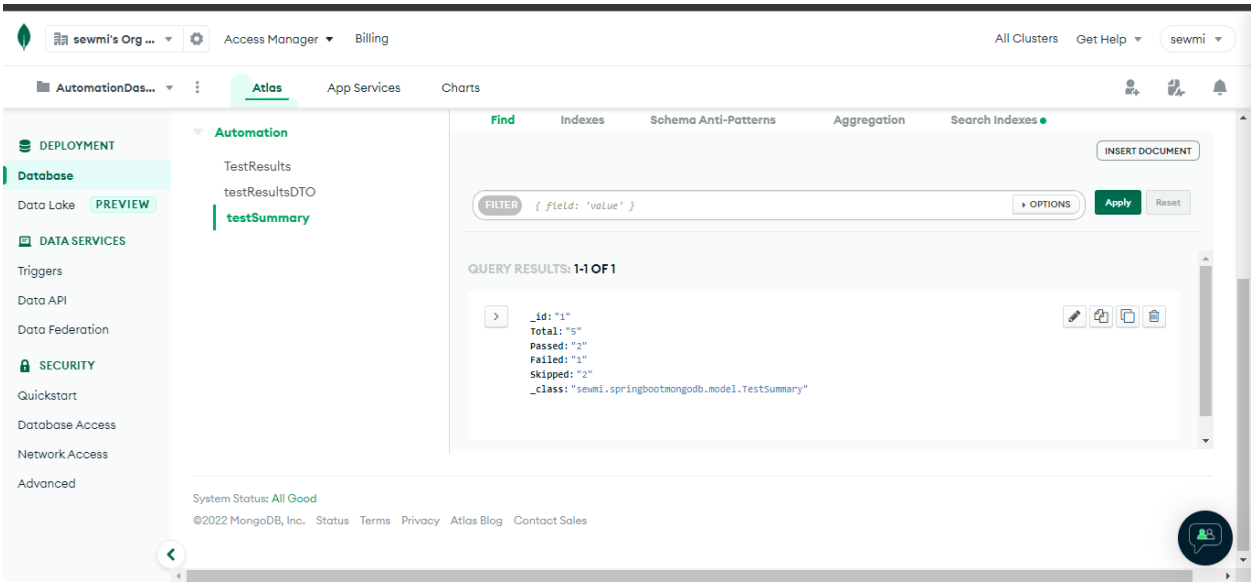

Figure 7.4.27 Data Collections – test summary data in cloud mongo db atlas

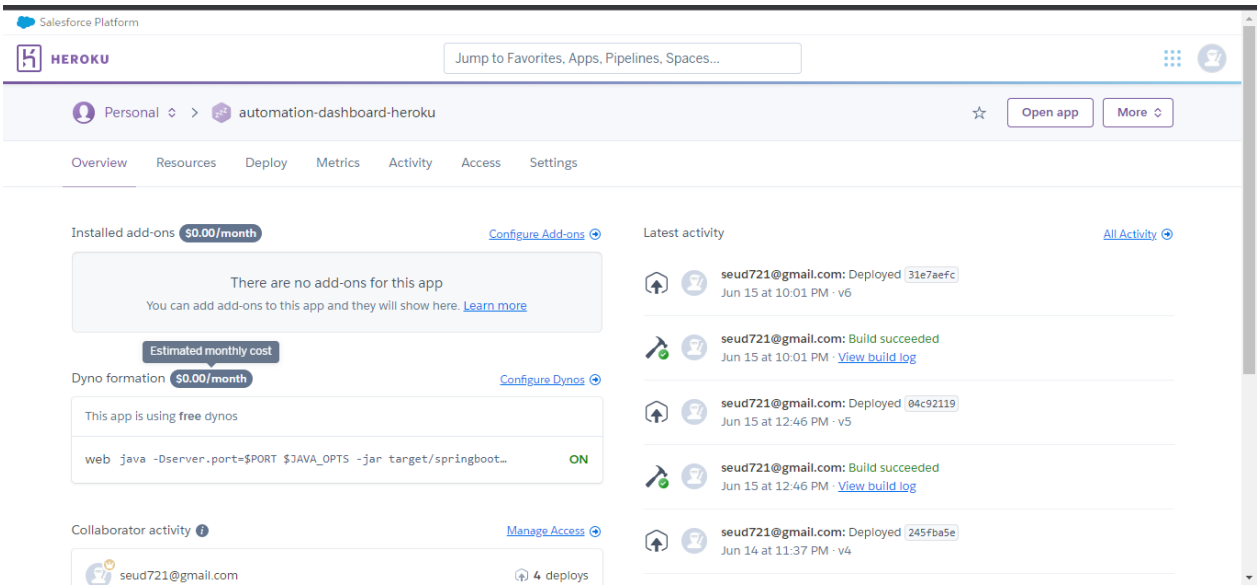

Figure 7.4.28 Heroku – Rest API Manager deployed to heroku

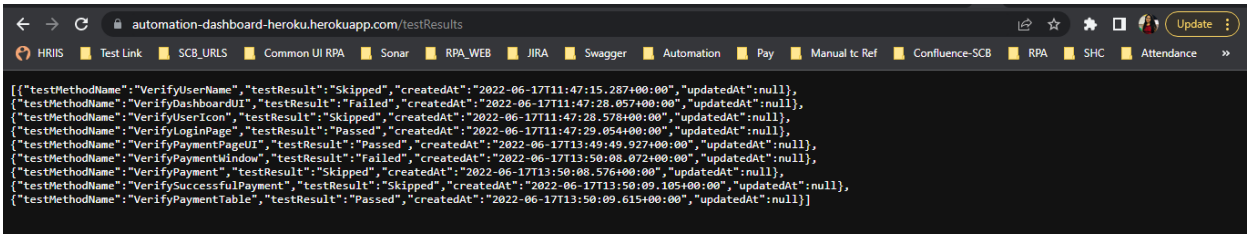

Figure 7.4.29 Heroku – Rest API Manager Data in hosted URL

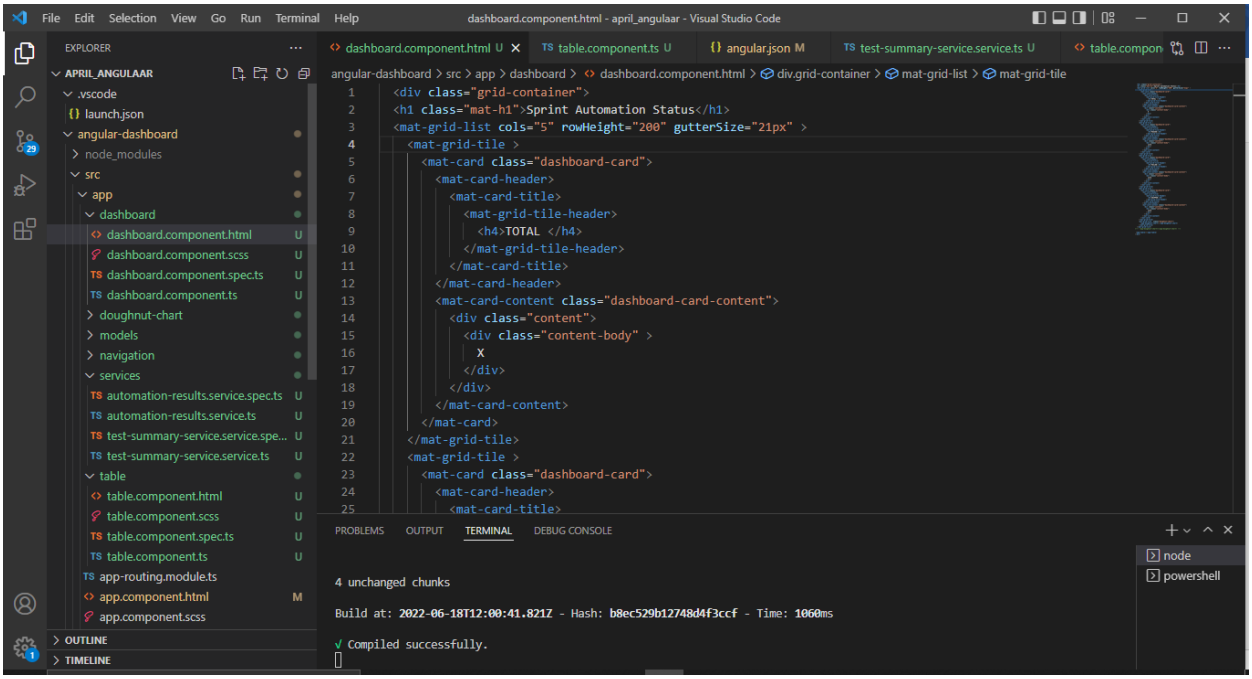

Figure 7.4.30 Angular front end development- dashboard component html

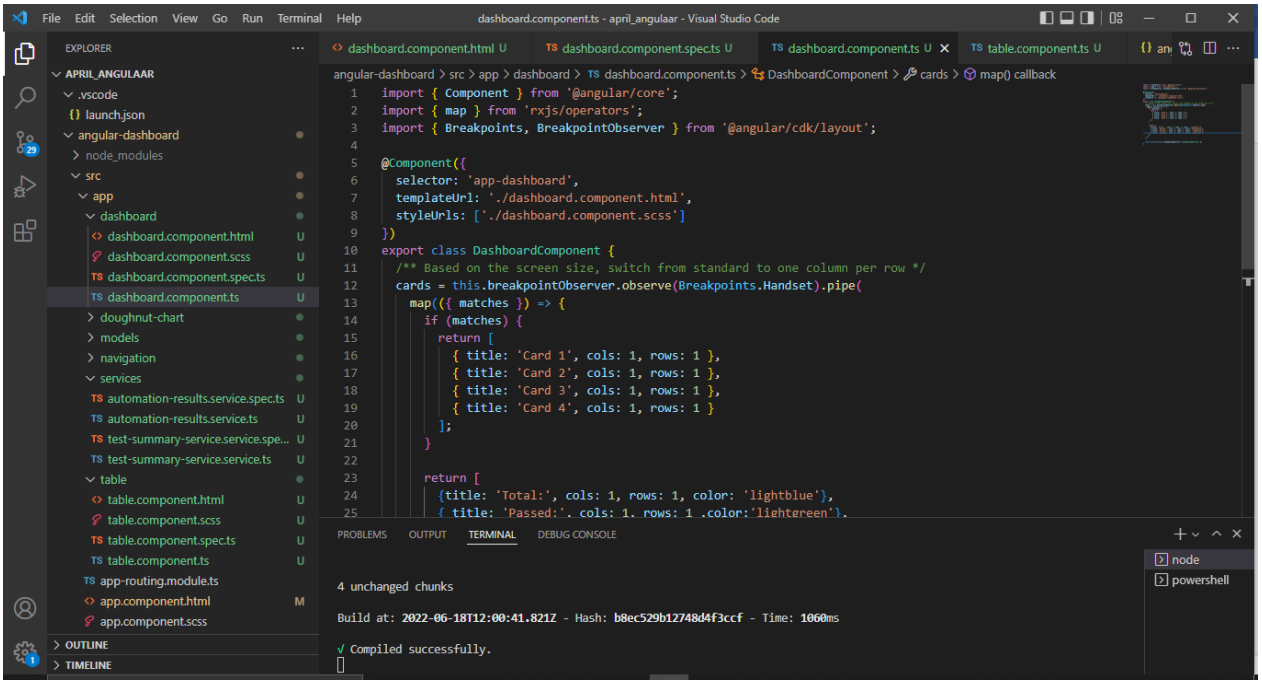

Figure 7.4.31 Angular front end development- dashboard component ts

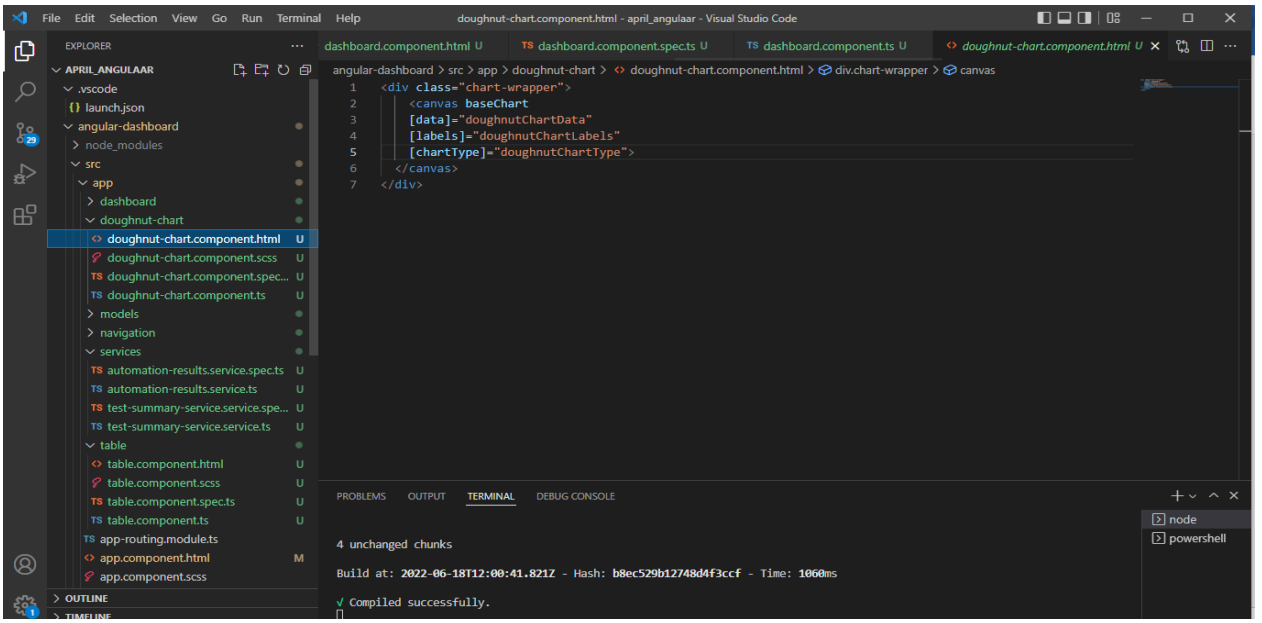

Figure 7.4.32 Angular front end development- Doughnut chart component html

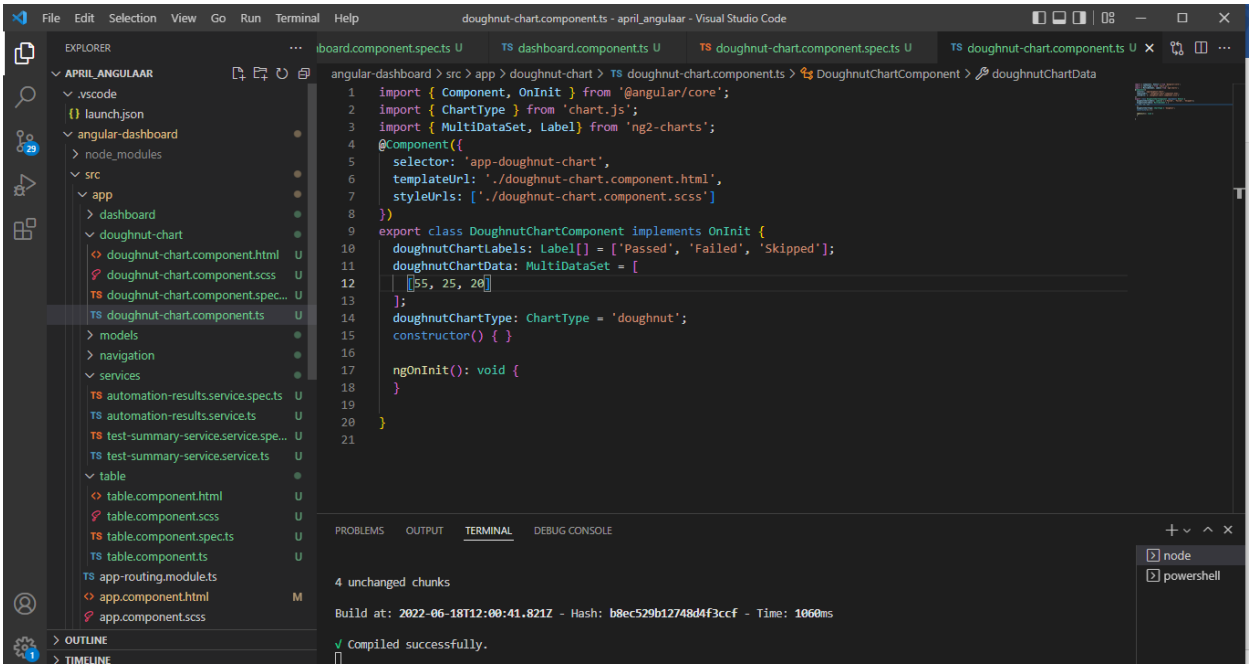

Figure 7.4.33 Angular front end development- Doughnut chart component ts

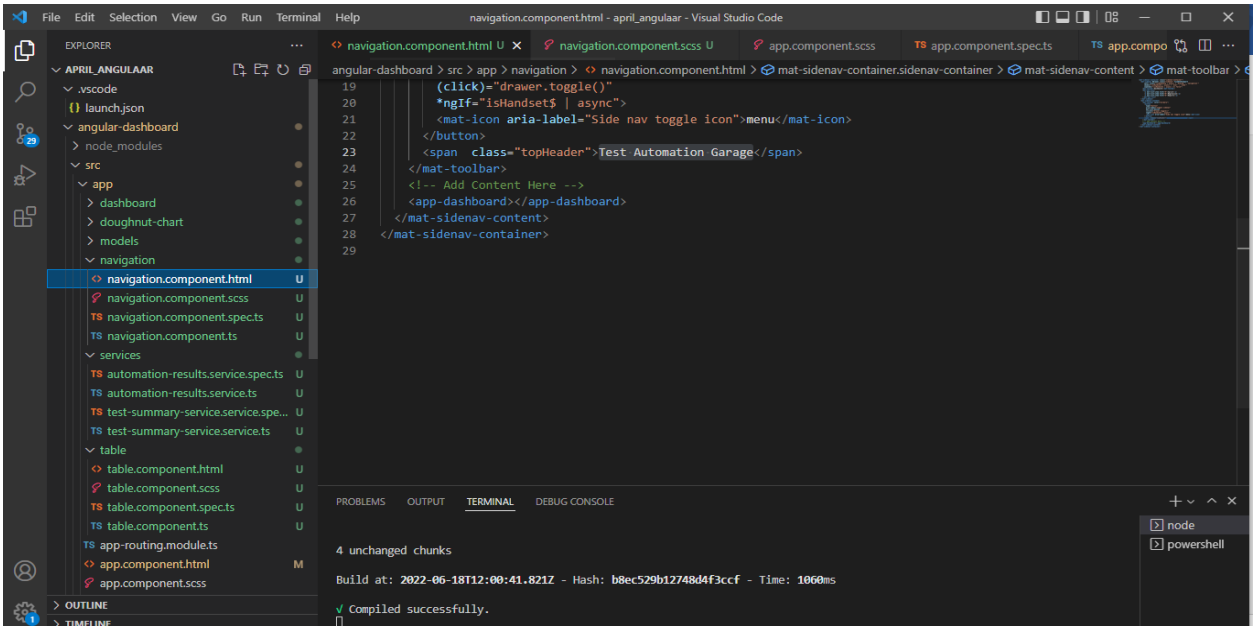

Figure 7.4.34 Angular front end development- Navigation component html

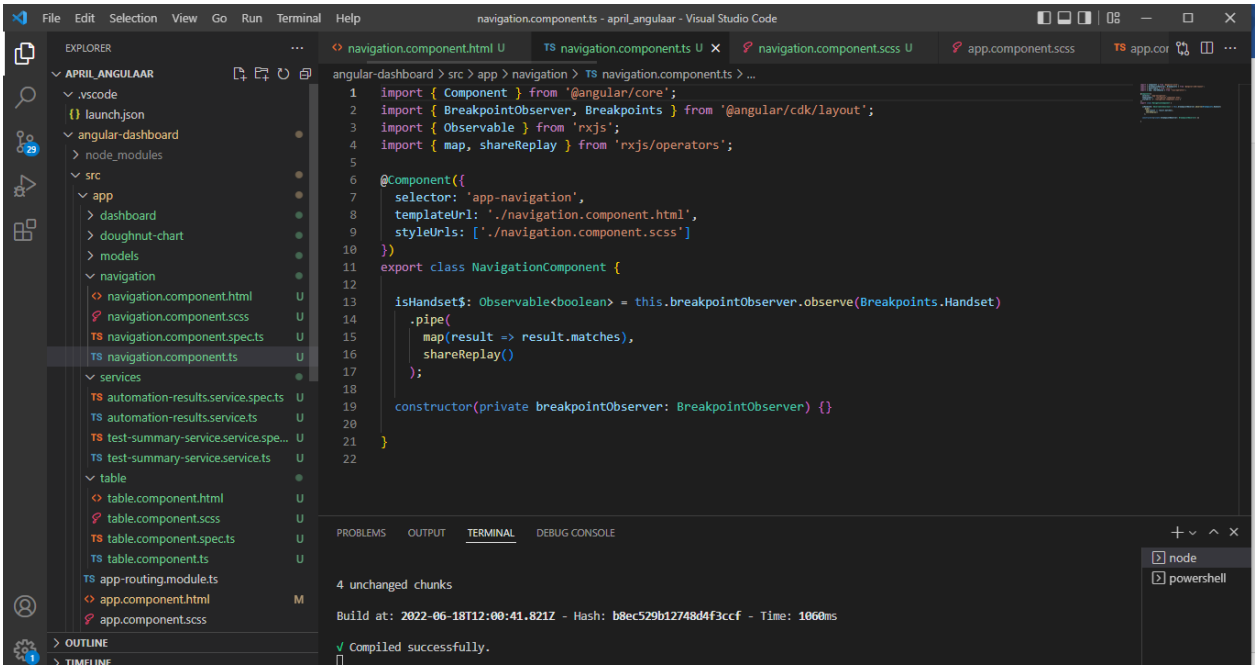

Figure 7.4.35 Angular front end development- Navigation component ts

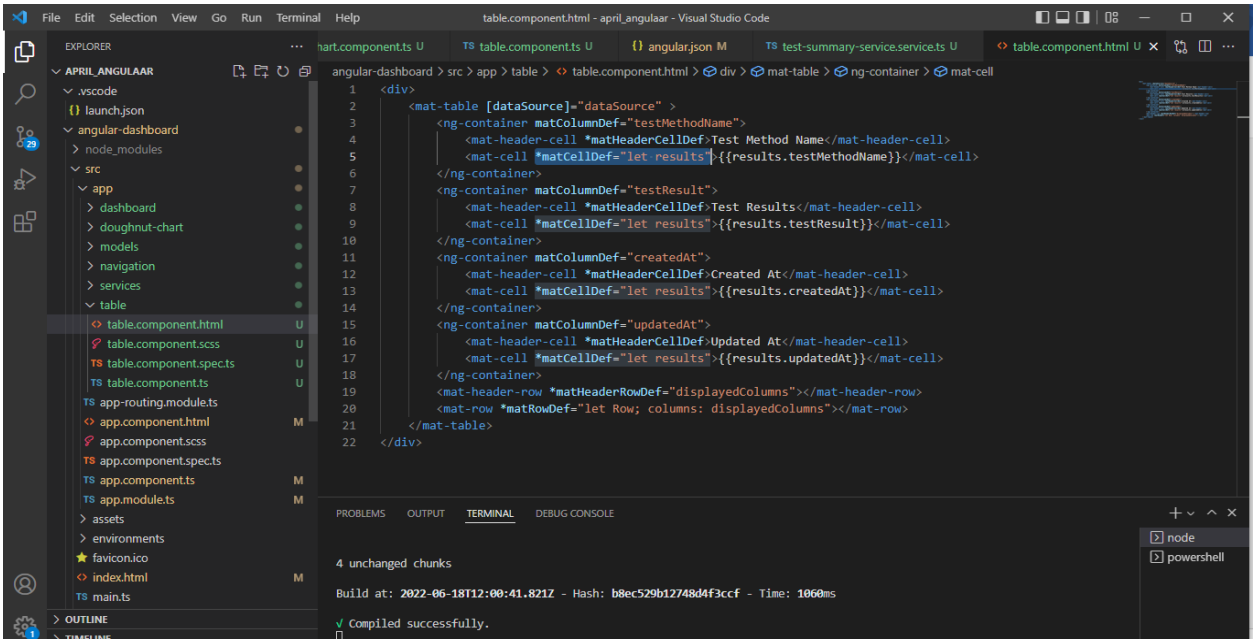

Figure 7.4.36 Angular front end development- Table component html

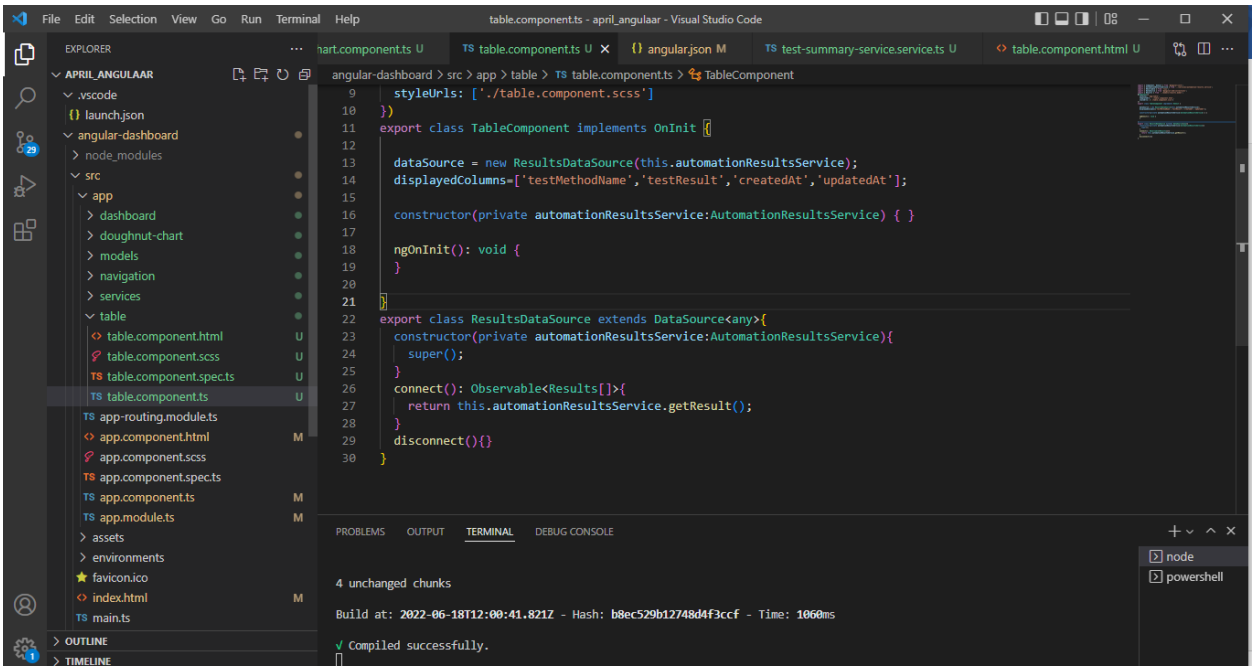

Figure 7.4.37 Angular front end development- Table component ts

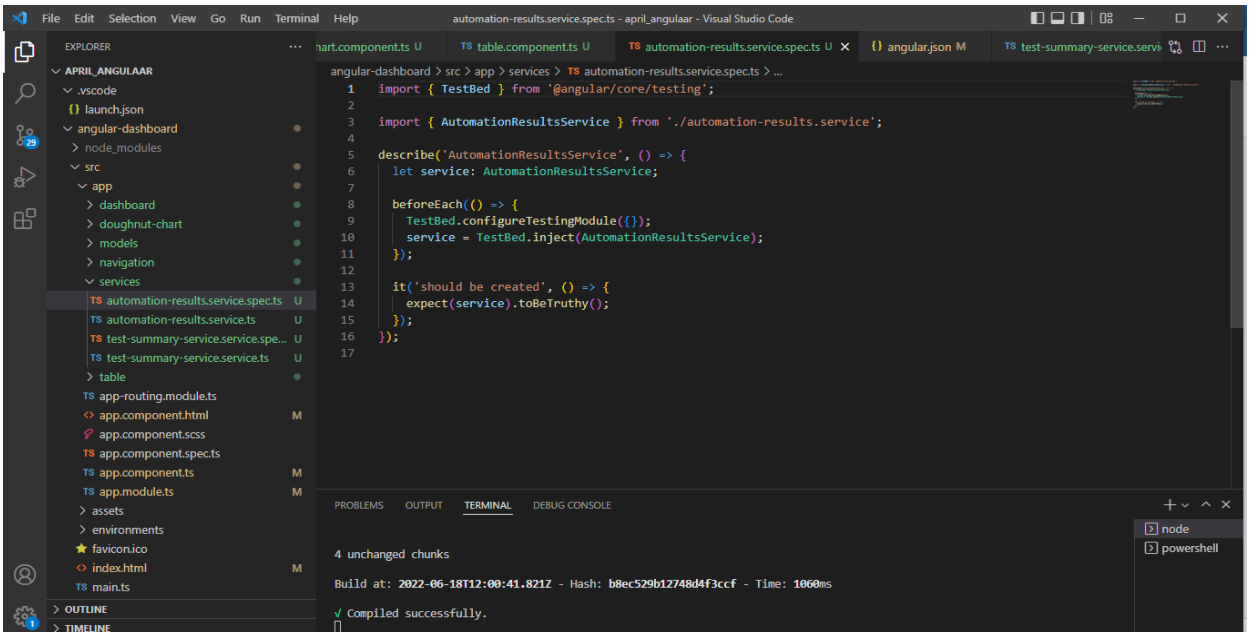

Figure 7.4.38 Angular front end development- Services for automation results spec ts

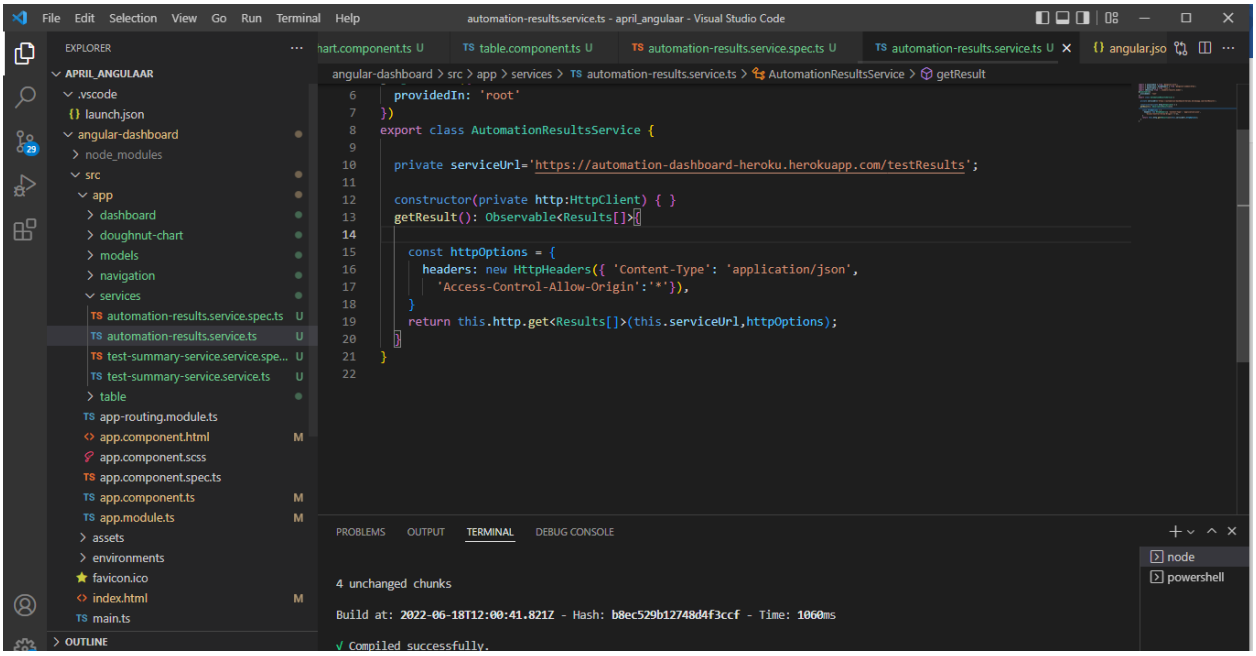

Figure 7.4.39 Angular front end development- Services for automation results service ts

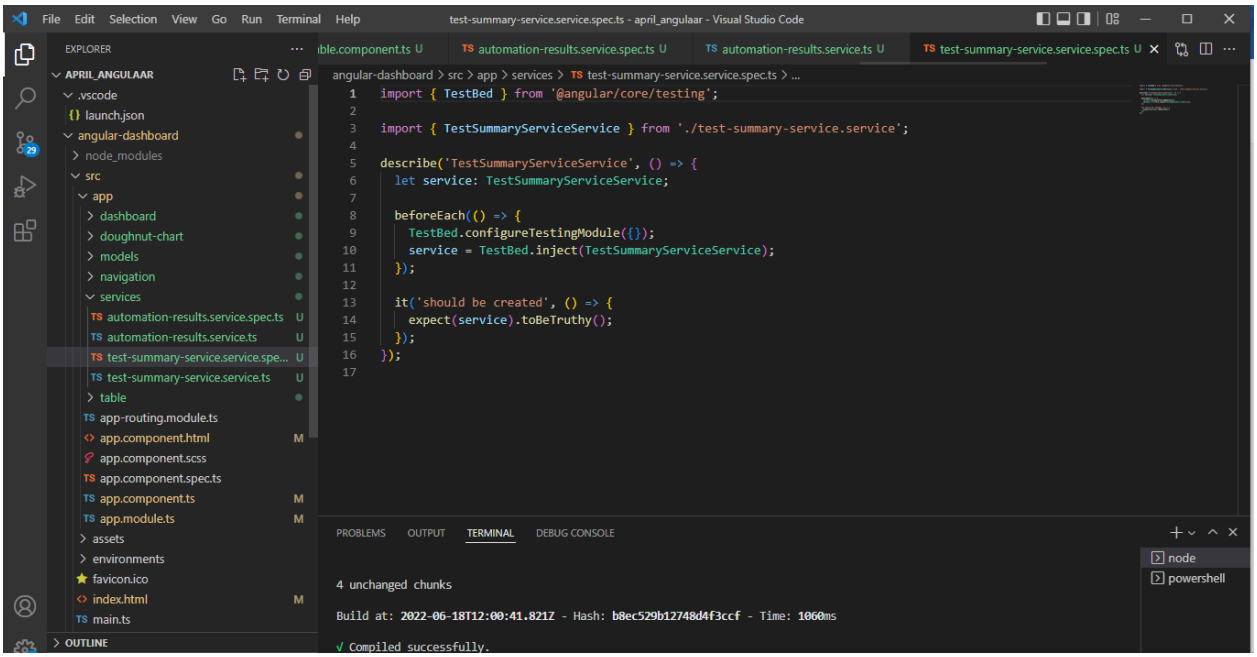

Figure 7.4.40 Angular front end development- Services for automation summary spec ts

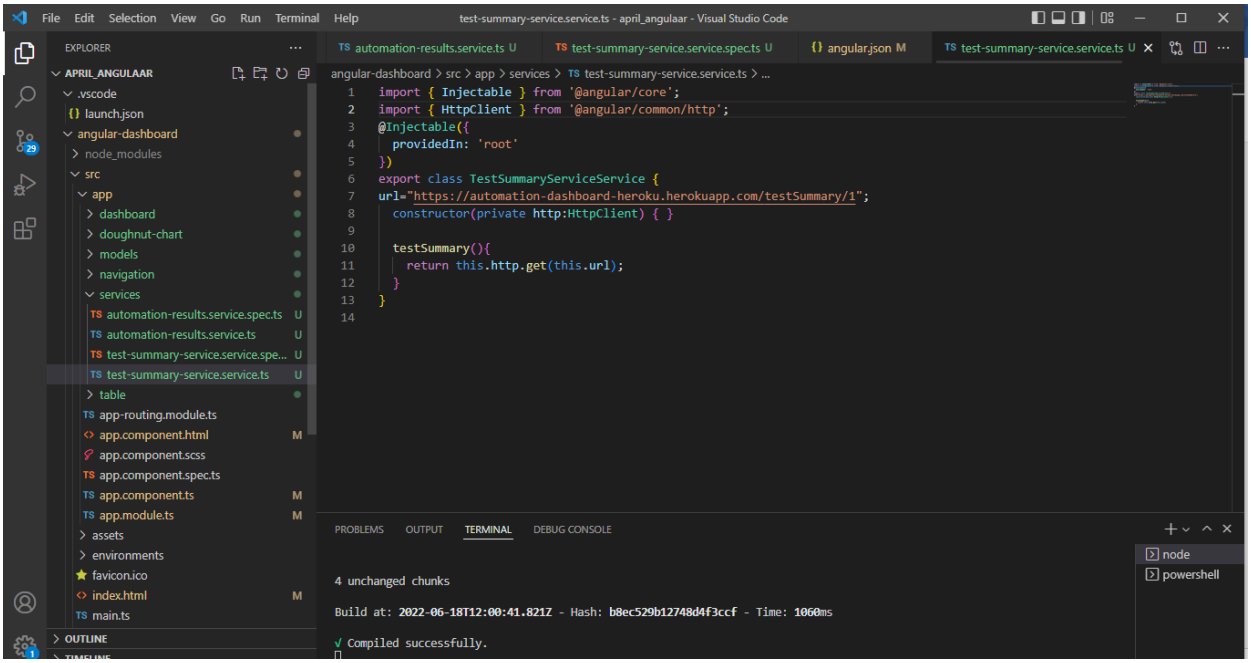

Figure 7.4.41 Angular front end development- Services for automation summary ts

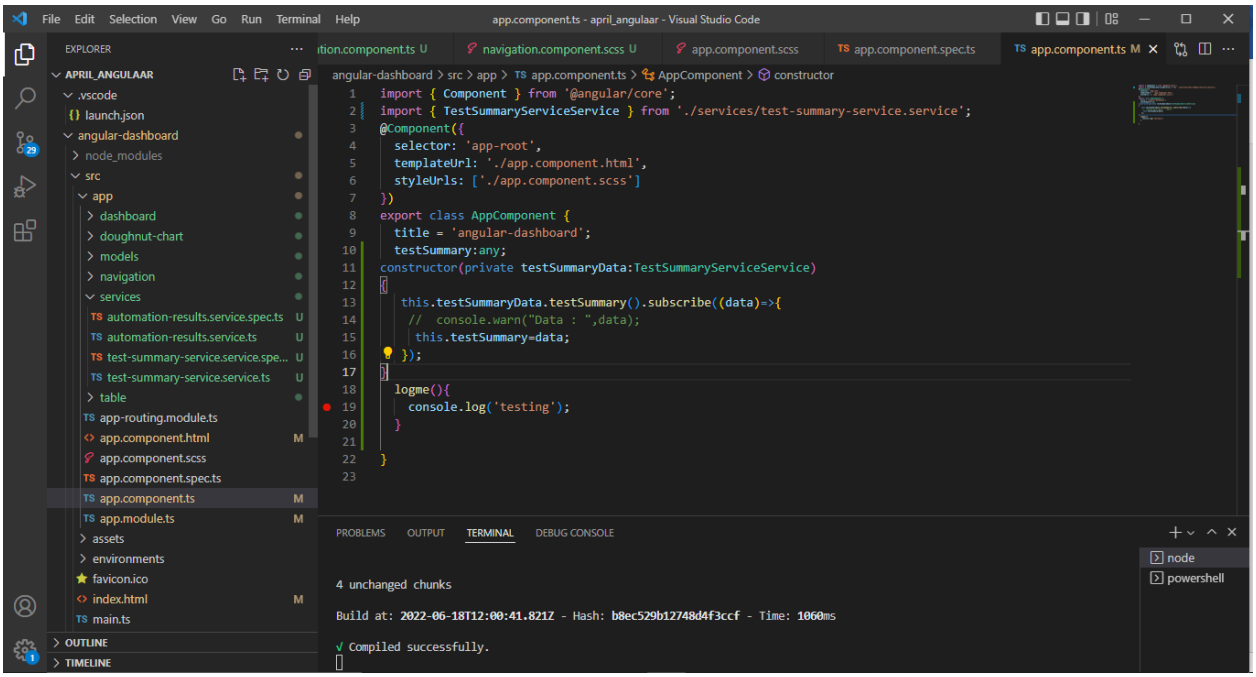

Figure 7.4.42 Angular front end development- app component ts

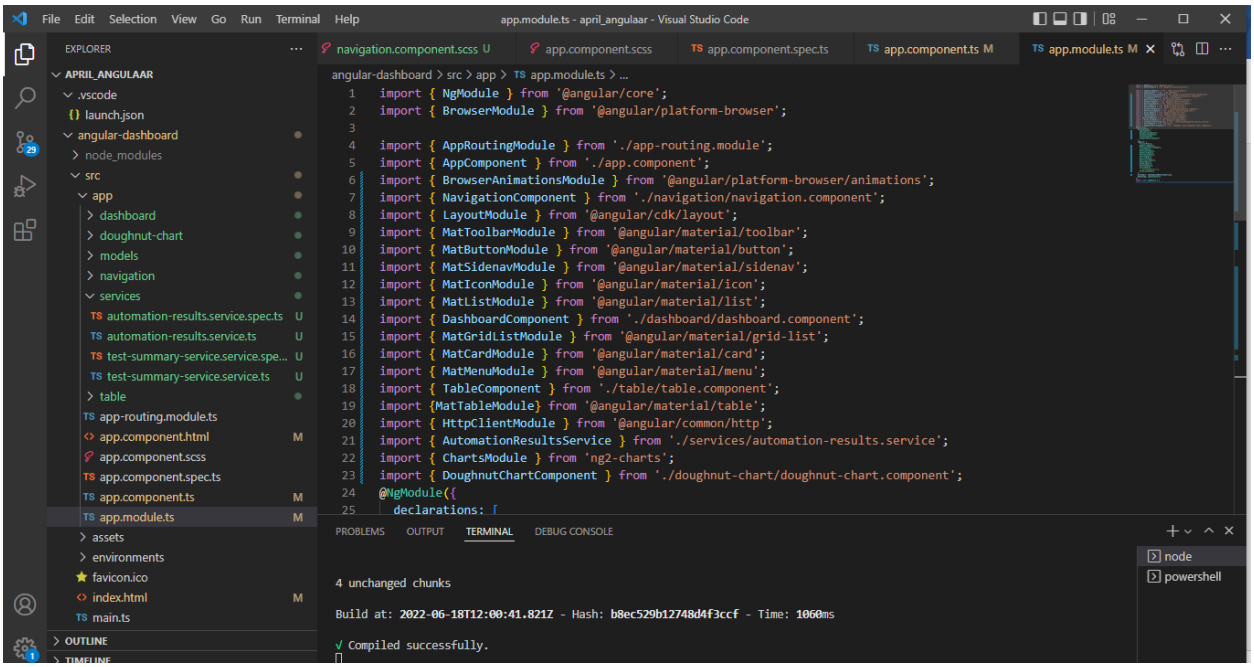

Figure7.4.43 Angular front end development- app module ts 01

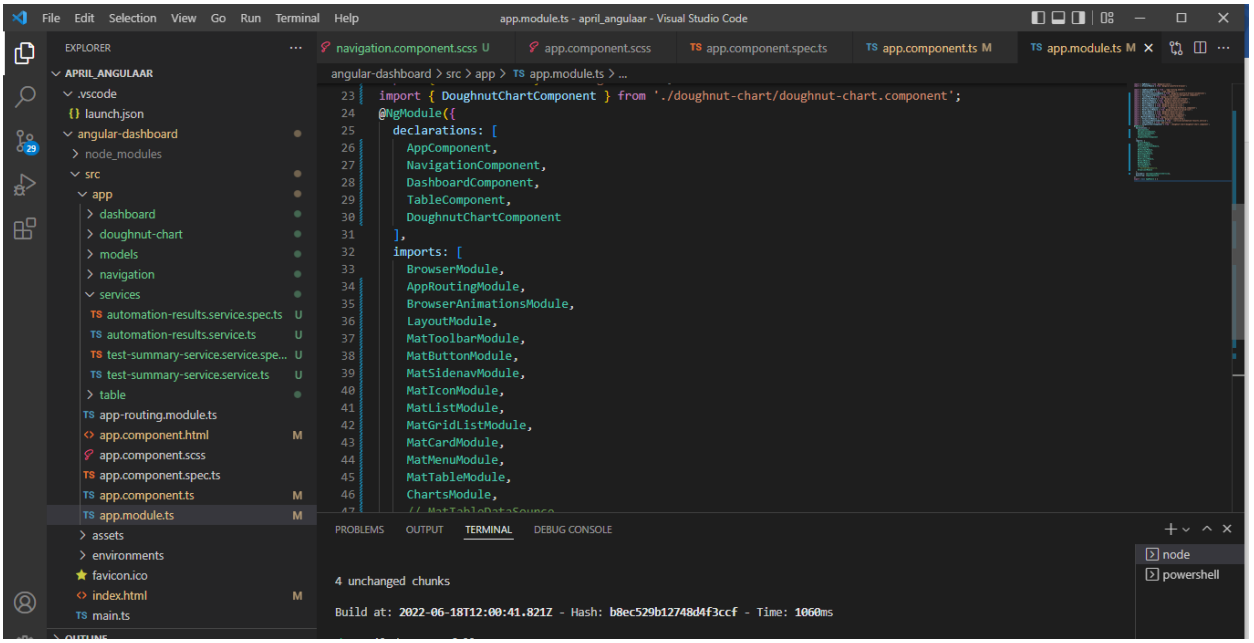

Figure 7.4.44 Angular front end development- app module ts 02

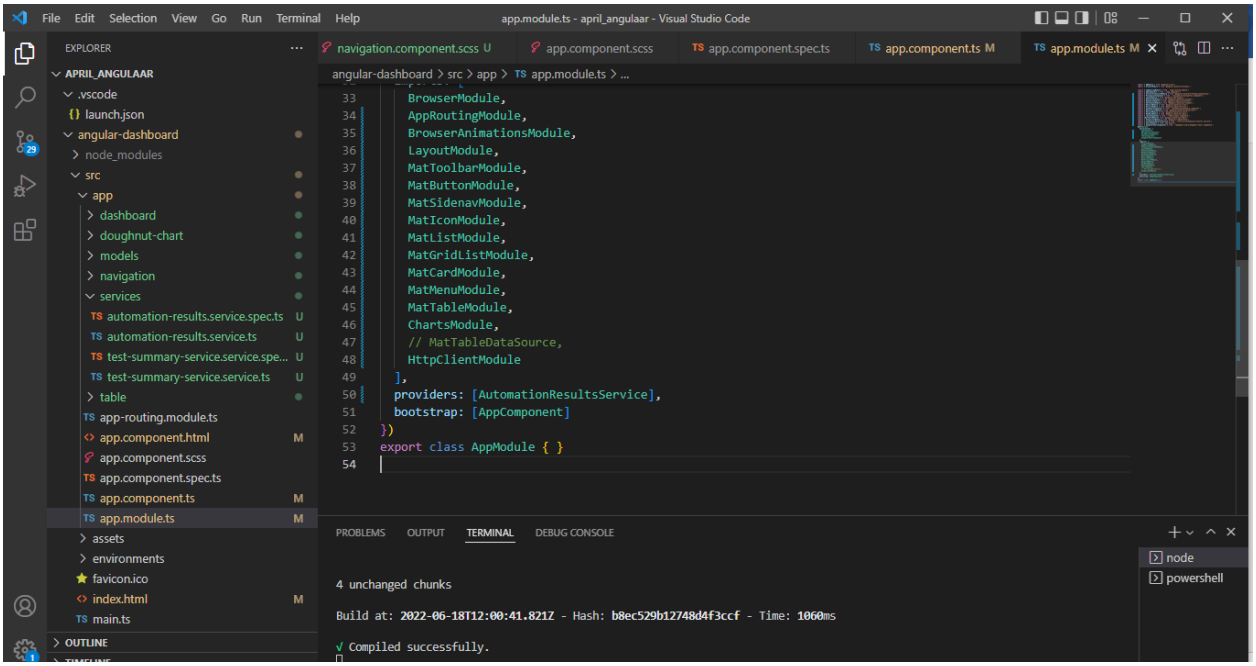

Figure 7.4.45 Angular front end development- app module ts 03

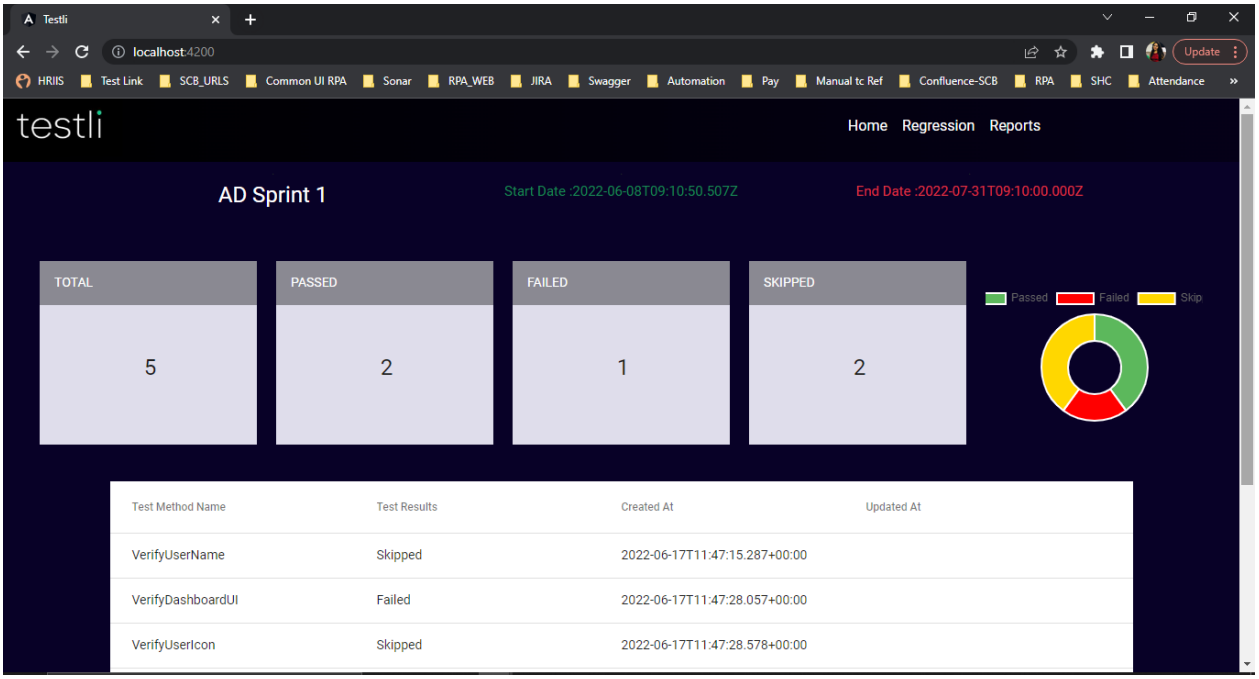

Figure 7.4.46 Angular front end development- Hosted Decision making Dashboard