FRAMEWORK TO MIGRATE ANGULARJS BASED LEGACY WEB APPLICATION TO REACT COMPONENT ARCHITECTURE

Thilanka Kaushalya Lanka Geeganage

179328A

Degree of Master of Science

Department of Computer Science and Engineering

University of Moratuwa
Sri Lanka

March 2021

FRAMEWORK TO MIGRATE ANGULARJS BASED LEGACY WEB APPLICATION TO REACT COMPONENT ARCHITECTURE

Thilanka Kaushalya Lanka Geeganage

179328A

Thesis submitted in partial fulfillment of the requirements for the degree Master of Science

Department of Computer Science and Engineering

University of Moratuwa
Sri Lanka

March 2021

DECLARATION

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for 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.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: .... Date:...................

Name: Thilanka Kaushalya Lanka Geeganage

I certify that the declaration above by the candidate is true to the best of my knowledge and that this report is acceptable for evaluation for the Masters thesis.

Signature of the supervisor: .... Date:...................

Name: Dr. Indika Perera

ABSTRACT

Due to the increased complexity and the high expectation of the modern web applications, the legacy applications built on top of previous generation frontend frameworks such as AngularJS fail to serve both end users and developers. The AngularJS based legacy applications start poorly performing when the application grows larger. At the same time they become highly challenging for developers to maintain and add new functionality over time. The next generation component based frontend frameworks such as React has outperformed AngularJS from both the end user and developer expectations. Since the whole industry has evolved to gain the benefits of the component architecture, sooner or later all these legacy applications need to migrate to new component based architecture. The concepts, patterns and the architecture of the old school AngularJS application and the component based React applications are different so the migration is not a straightforward step. The successor of AngularJS which is named as Angular2 provides some tooling to support migrating old AngularJS applications to Angular2 but those are not directly compatible with React migration. This makes the organizations and developers who are planning to migrate AngularJS application to stick with Angular2 which is a less popular and flexible framework when compared with React. This research tries to define a clear migration path from AngularJS applications to React applications which will allow organizations and developers to choose React also as a goto framework option and will save hundreds of research and development hours in setting up the migration.

This research compares and contrasts approaches of legacy frontend application migrations with respect to steps, best practices, technologies tools and pain points. By comparing different approaches, the research provides a step by step guideline which can be referred to and followed when migrating any AngularJS based legacy application to component base React application. These guidelines are organized as a migration framework with steps of refactorings and provide as the first outcome of this research. As the second step of this research, a migration assistant tool which is called as Ng-React Copilot was implemented to guide the developer through the migration steps and apply them to the existing code base. The tool was developed by converting the critical refactorings proposed in the framework into a set of detection algorithms and providing the ability to scan against the provided codebase as a command line tool as well as an integrated tool with populer IDEs.

The framework and the tool were validated by applying them to selected small, medium and enterprise level AngularJS legacy applications. One application was selected from each category and used the tool and the framework to step by step migrating to React where the small application was fully migrated to React while the medium and enterprise applications were migrated partially with the interest of the time. The results from the migration assistant tool and the framework were collected and validated the accuracy of them and except for a few false positives, the migration assistant tool was detecting the required refactoring accurately and was making the migration straight forward. Improving the algorithms and configurations to avoid false positives, improve the tool to guide the users as a step by step workflow rather than showing all the refactorings at once, automating few more time consuming manual refactorings, improving the framework to address external library coupling and improve the tool to guide the user on where in the application to start the migration were identified as future improvements for the identified limitations during the evaluation process. By looking at the evaluation results, the migration framework and the Ng-React Copilot tool can be considered as good starting materials to get assistance for AngularJS to React application migration and the effectiveness can be further improved by applying the identified future improvements.

ACKNOWLEDGEMENTS

I would like to express my gratitude to Dr. Indika Perera, my project supervisor for providing advice and guiding the project into success. Even though I had an idea about the domain initially I didn’t have a clear idea on how to converge the scope of the research into a meaningful outcome. Dr. Indika helped me and guided on how to narrow down the scope and focusing on building up the framework proposed in this report. Also I would like to express my gratitude to my family for supporting me in focusing on the research. Last but not least I wish to thankful for all my colleagues who have encouraged me to continue the research by expressing their experiences.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

DOM	Document Object Model
MVC	Model View Controller
MVVM	Model View View Model
MVP	Model View Presenter
MV*	Model View Any
AST	Abstract Syntax Tree
JS	Javascript
CSS	Cascading Style Sheets
XML	Extensible Markup Language
JSX	Javascript XML