Rule Based Content Player

Submitted in fulfillment of the requirements for the Degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

By

Akash Patel

15BIT029

Nikita Dandwani

15BCE075

Guided By

Prof. Usha Patel

Prof. Meenaxi Tank

Computer Science and Engineering Department

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Ahmedabad 382481

Rule Based Content Player

BY
AKASH PATEL
NIKITA DANDWANI

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

Ahmedabad 382481

Rule Based Content Player

Major Project Report

Submitted in fulfillment of the requirements for the Degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

By

Akash Patel
15BIT029

Nikita Dandwani
15BCE075

Institute Guides
Prof. Usha Patel,
Prof. Meenaxi Tank,
Assistant Professor,
Department of Computer science,
Nirma University

Industry Guides
Mr. Padmanabhan A,
Mrs. Sanjana Hans,
Mrs. Akhaya Raghunath,
Mr. Shreekhar R,
ETA Department
Infosys Ltd.

Department of Computer Science and Engineering
Institute of Technology
NIRMA UNIVERSITY
Ahmedabad 382481

May 2018

Infosys | Education, Training and Assessment
Navigate your next

CERTIFICATE OF

COMPLETION

OF INTERNSHIP

This is to certify that

Akash Ashwinbhai Patel

From

Nirma Institute Of Technology

has successfully completed the final semester project and internship programme during

January - May, 2019

Satheesha B Nanjappa
Vice President and Head, Global Education Center

TO WHOMSOEVER IT MAY CONCERN

This is to certify that Akash Patel (15BIT029), a student of B.Tech. in Information Technology from Institute of Technology, Nirma University worked in Infosys Ltd. as a project trainee during 14-Jan- '19 to 7-May- '19 of dissertation. During this period, he was found regular and had done his project on "Rule Based Content Player", under my supervision.

He has worked with utmost dedication and high level of engineering and analytical competence.

We wish him all the best for his future endeavors.

Date:

Akshaya Raghunath
Senior Associate (ETA)

Prof. Usha Patel

Undertaking for Originality of the Work

I, Akash Patel, Roll No.15BIT029, give undertaking that the Major Project entitled “Rule Based Content Player” submitted by me, towards the partial fulfillment of the requirements for the degree of Bachelor of Technology in Information Technology of Nirma University, Ahmedabad, is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. I understand that in the event of any similarity found subsequently with any other published work or any project report elsewhere; it will result in severe disciplinary action.

Signature of Student

Date:

Place:

Endorsed by:

Akshaya Raghunath

Prof. Usha Patel

CERTIFICATE

This is to certify that the Major Project Report entitled “Rule Based Content Player” submitted by Akash Patel(15BIT029) towards the partial fulfilment of the requirements for the award of degree in Bachelor of Technology in the field of Information Technology of Nirma University is the record of work carried out by him under our supervision and guidance. The work submitted has in our opinion reached a level required for being accepted for examination. The results embodied in this major project work to the best of our knowledge have not been submitted to any other University or Institution for award of any degree or diploma.

Date:

Industry – Guide
Akshaya Reghunath
Senior Associate
Infosys Limited.

Institute – Guide
Prof. Usha Patel
Assistant Professor
Nirma University

Head of Department
Dr. Madhuri D Bhavsar
Department of Computer Science & Engineering
Nirma University
Ahmedabad

Director
Dr. Alka Mahajan
Institute of Technology
Nirma University
Ahmedabad

ACKNOWLEDGEMENT

We would like to acknowledge the sincere efforts put in by all the people who helped in completing this project.

We express our gratitude to our mentors Mr. Padmanabhan A, Mrs. Sanjna Hans, Mrs. Akshaya Raghunath and Mr. Shreekhar R for their constant support, motivation and guidance throughout the project.

We are profoundly obliged to Prof. Usha Patel and Prof. Meenaxi Tank for their direction and steady supervision to provide essential data and support in completing the project.

We want to express our gratitude towards our parents, friends and individual from Institute of Technology, Nirma University for their kind co-operation and support which helped us in fulfillment of this project.

ABSTRACT

Rule Based Content Player is a dynamic player in Angular 7 that displays the learning material and resources based on the user's previous knowledge of the course selected from the list of available courses.

The basic aim of the system is to create personalized learning application adaptive to the learner's knowledge. The learning material available for the course is also designed in such a way that only the required contents are available to user. The Rules which guide how the course path for the particular learner should be made are supplied in a particular format to the system.

Modern learning apps overlooks the fact what the user might know relevant to that particular course but this system segregates the available content based on the rules defined for that particular course. This application provides a best user experience so that they can learn their course in a better way.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 THE SYSTEM

1.1.1 DEFINITION OF THE SYSTEM

Rule Based Content Player is an Angular based application that displays the learning resources to the learner in a personalized manner. The system uses a set of rules that checks what previous knowledge is there with the learner and what are the prerequisites of the courses. The application uses the JSON file as a database for backend logic implementation.

1.1.2 OBJECTIVES

Our Objective is to make a content player that plays all type of resource in a single player such as videos, images, interactions, assessments etc. It should work across different levels of users and they can learn their content relevant to them. For example, there are two type of users let say Manager and Developer. If the user is a Manager, then he only wants the description about how it works rather than the coding so the player should display the content that relevant to the manager. In the same way Developer should get the content related to implementation.

Users can respond to various queries asked by the system during the learning. Also he can take assessment to improve his knowledge level.

1.1.3 PROPOSED SYSTEM

The proposed system is a rule based system. It works based on the predefined rules that provided by the educator during the creation of the course. The educator provides the rules of a particular course in JSON format. The system plays the content by evaluating the rule provided in the JSON file. The system provides queries to the learner to know the basic knowledge level of the learner. The Questions and other resources such as Videos, Images etc. also provided in to the system in the form of JSON by the educator. Based on the learner's interaction to the questions the system evaluates appropriate rule and plays the content in the player. The learner can take assessment so that he can check their knowledge level. Content player is capable of handling all type of file i.e. Videos, Images, Questions, Assessment etc. The system provides responsive UI which helps to work in all type of devices.

1.2 PROJECT PROFILE

1.2.1 PROJECT TITLE

RULE BASED CONTENT PLAYER
Create a better learning platform which shows the course based on the rules and resources decide by the educator.

1.2.2 SCOPE OF PROJECT

The main goal of our project is to develop a player that can show the content that relevant to the learner based on the rules. Rules are defined in a json file. The rule is defined by the educator who creates the course. Rule defines the flow of the course.

1.2.3 PROJECT TEAM

We completed our project successfully under the supervision of our mentor. Our team has 3 members namely

1. Akash Patel
2. Nikita Dandwani
3. Sooraj Soman

The initial research work for tools and technologies to be used was done by every member of the team. The documentation and agile sheet is divided equally to every member of the team. Our project is divided into different components. Nikita work on the Content Player component where Akash and Sooraj worked on the Rule Evaluator, Master Engine, Resource Meta Data Reader, Variable, Resource, Rule, Course-List Component. Integration of these components is done together.

2. SYSTEM ANALYSIS

2.1 FEASIBILITY ANALYSIS

2.1.1 OPERATIONAL FEASIBILITY

The proposed system is a kind of abstraction that can be used for different levels of user based on his knowledge level. The content will display based on the knowledge level of the learner.

2.1.2 TECHNICAL FEASIBILITY

The system is developed in Angular 7 and it uses JSON file as back end so that system works in small end systems also. It needs a web browser preferably Google Chrome to run the system completely. System is work in the client side so that the performance may vary according to the client machines. System takes some memory to save the learning resources in the client side.

2.1.3 FINANCIAL AND ECONOMIC FEASIBILITY

The system is quite feasible and affordable in economic terms. The hardware and software requirements do not cost much. The system does not require any additional hardware setup and software used with which application is developed is Open source therefore no additional cost is added.

2.2 REQUIREMENT ANALYSIS

The requirement analysis process is carried out by arranging brainstorming session where product owner, domain expert, scrum master and the development team are present. During this session, following requirements is collected.

2.2.1 FUNCTIONAL REQUIREMENTS

The Functional requirements include the Business rules of the system and the behavior of the system. Functional requirements define the logic of the system and the way it is supposed to operate.

US01: As a learner, I should be able to view the list of available courses so that I can choose a particular course for learning.

US02: As a learner, I should be able to choose a course from the list of available courses so that I can start learning that specific course.

US03: As a learner, I should be able to view the starting content so that I can start the learning process for the chosen course.

US04: As a learner, I should be able to choose appropriate options for the queries raised during the course so that I can get course content relevant to me.

US05: As a learner, I should be able to submit the responses to the queries raised so that I can get course content relevant to me.

US06: As a learner, I should be able to view queries during the course so that I modify the course path.

US07: As a learner, I should be able to navigate to the next resource so that I can continue with the course.

US08: As a learner, I should be able to navigate to the previous resource so that I can refer to content that has already been displayed.

US09: As a learner, I should be able to get resources specific to the course that I have chosen so that there is no confusion with respect to the content displayed.

US10: As a learner, I should be shown the appropriate subsequent resource in the course after viewing the current one so that there is continuity in my learning.

US11: As a learner, I should be able to choose the options to the quizzes that are displayed during the course.

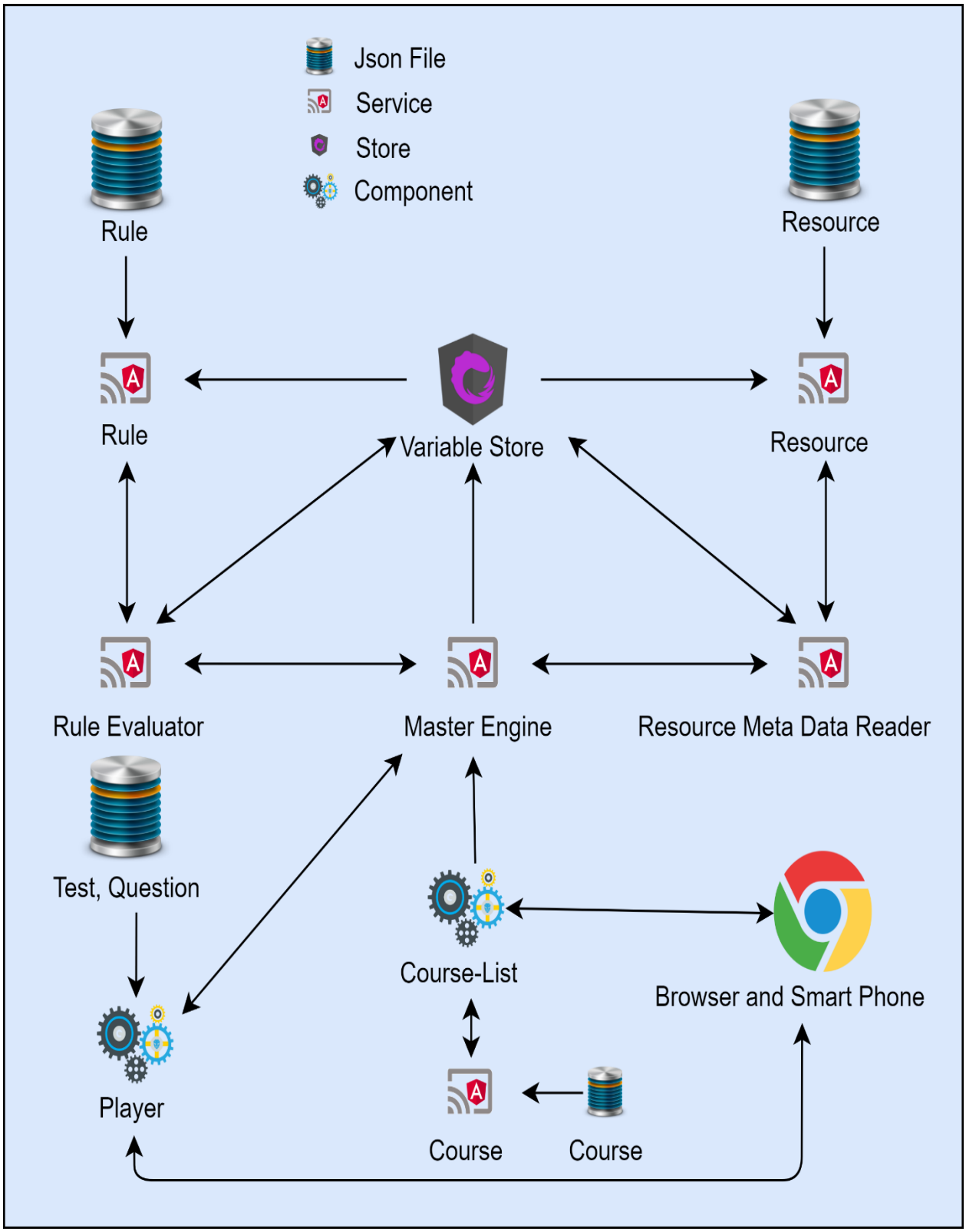
US12: As a learner, I should be able to choose the options to the assessment that are displayed during the course.

2.2.2 NON-FUNCTIONAL REQUIREMENTS

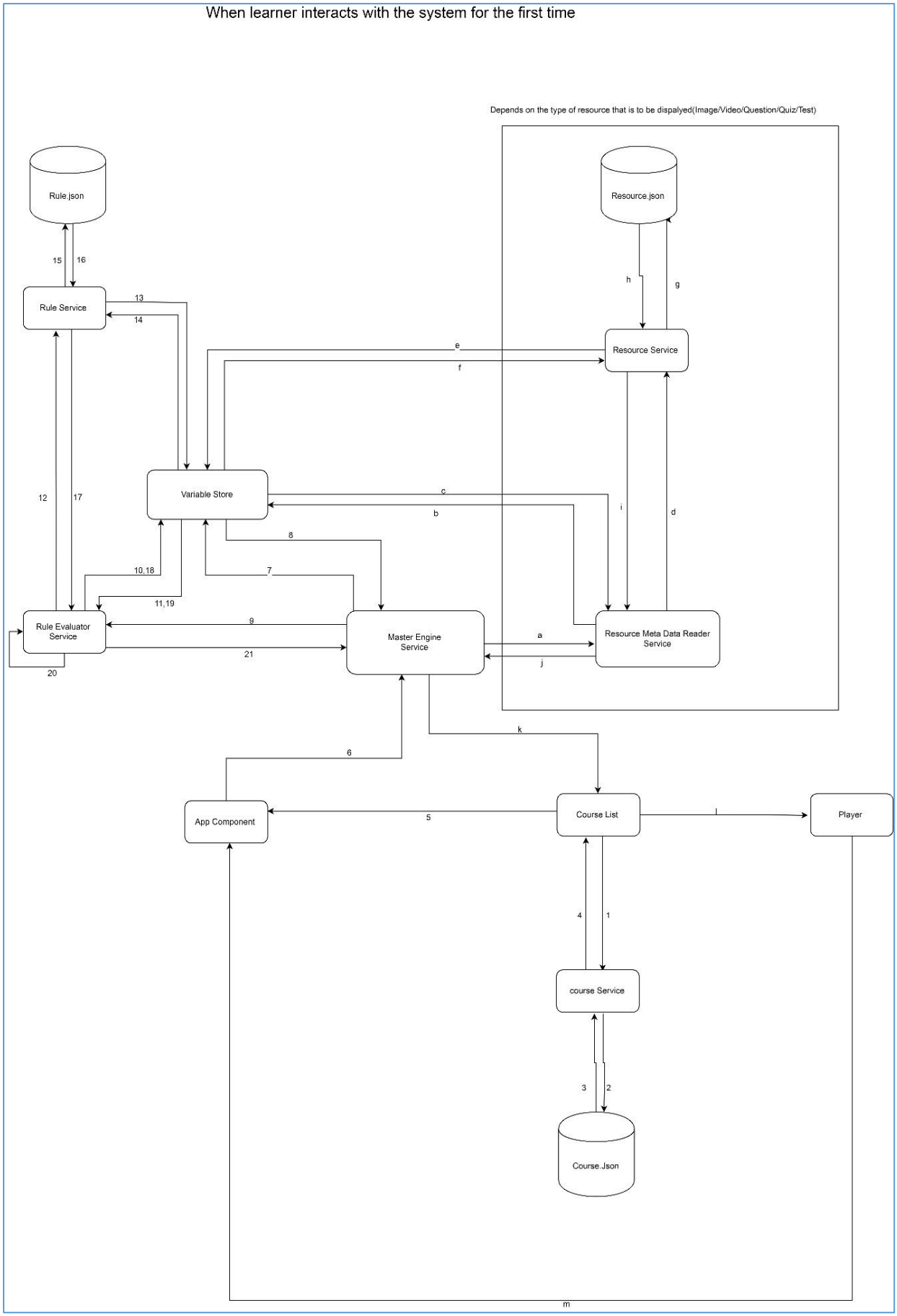
Non-Functional Requirements check the operational performance of the system that has nothing to do with the functional requirements of the system. It accounts for performance, scalability, Availability, Maintainability, Robustness of the system.
According to the above definition, our system is

1. Robust- Won't stop in case of system of failure.
2. Fault Tolerant- Display proper Error messages.
3. Performance- Response time is kept optimal so that all components are loaded at display time and system does not stops.
4. Scalable- Our system is highly scalable and can evaluate as many as rules given in json file.

2.3 FLOW DIAGRAM

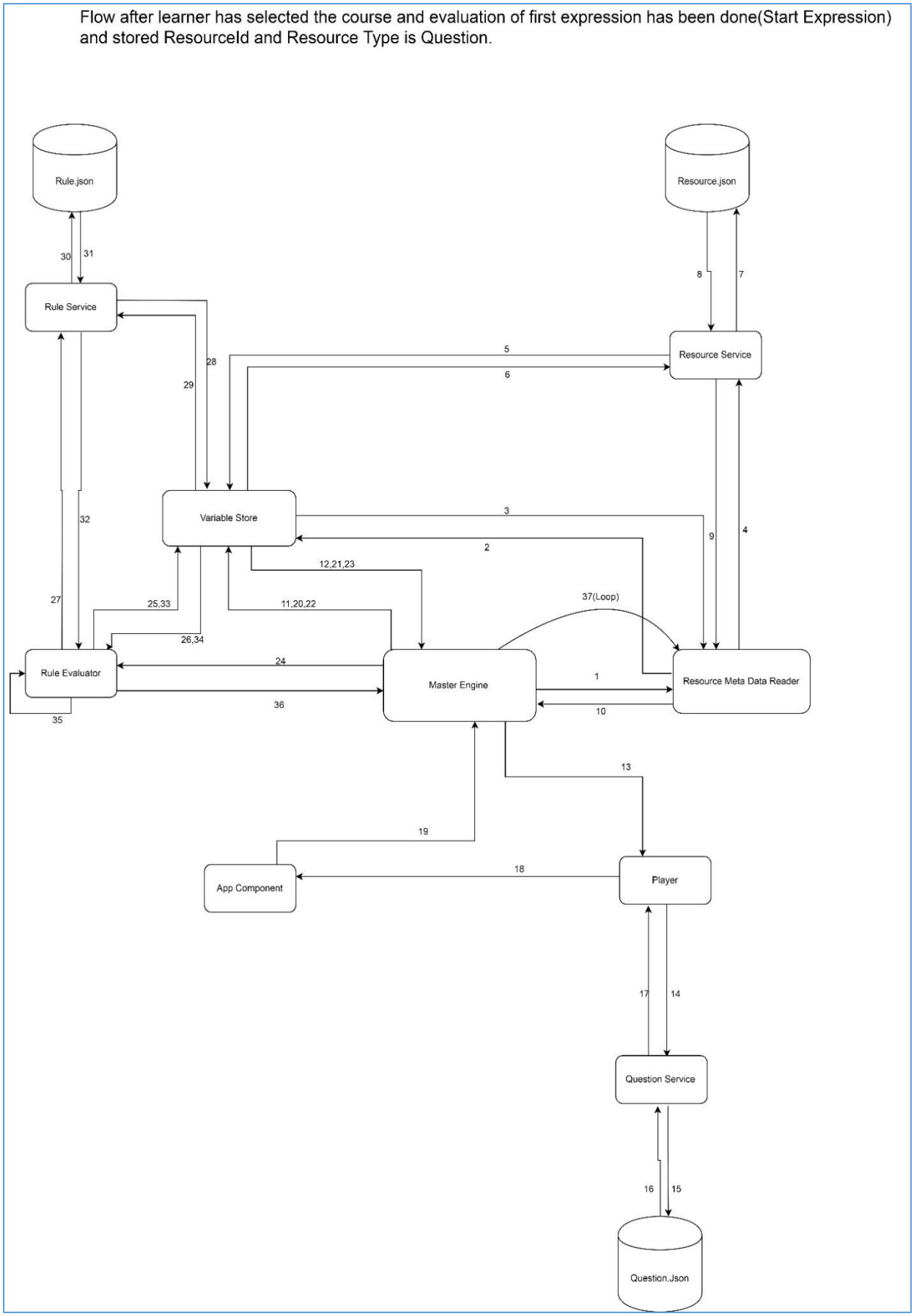


2.3.1 General Project Flow Diagram



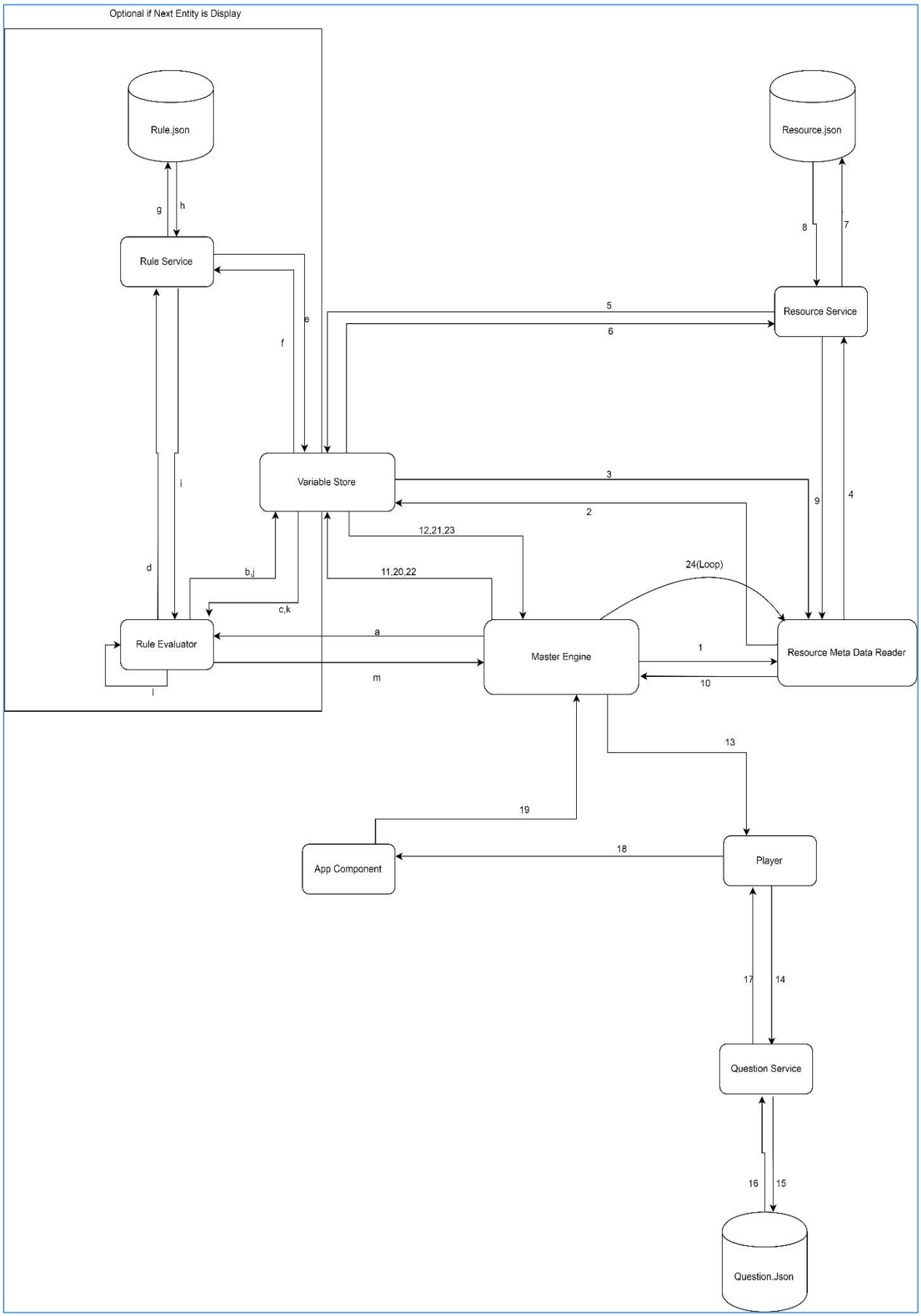
2.3.2 Flow Diagram for First Time Course Selection

Steps	Function
1	Course List component calls the Course Service
2,3	Course Service reads the the Course.Json to fetch the list of courses
4	Course service fetches the Course object and returns to Course List component
5	The list of all available courses are displayed on the Home Screen (App Component)
6	Selection of the Course (click event occurs) and Master Engine Service is called
7,8	Creating variable object with variable name (Course Name) and variable value (User selected course)and store object in variable box.
9	Master engine calls the rule Evaluator for getting first resource Id
10,11	Getting RuleId from store(Start expression)
12	Call Rule Service to get Rule Object using Rule Id.
13,14	Getting Rule Path from store
15,16	Rule Service reads Rule.json
17	Rule Service returns the Rule object to Rule Evaluator Service
18,19	Getting Variable value from Store
20	Evaluates the Rule if TrueAction or FalseAction is Evaluate Rule (Repeat from 10 to 19)
21	Returns the flow to Master Engine
a	Calling Resource Meta Reader Service function (getResourceTypeandLocation()).
b,c	Getting Resource Id from Store
d	Calling Resource Service to get the Resource Object.
e,f	Getting Resource Path from Store.
g,h	Reads Resource.Json
i	Resource Service returns the Resource Object using Resource Id.
j	Returns the flow to Master Engine
k	Flow goes to Course List Component
l,m	After the First expression evaluation, the first resource is displayed using Player component.



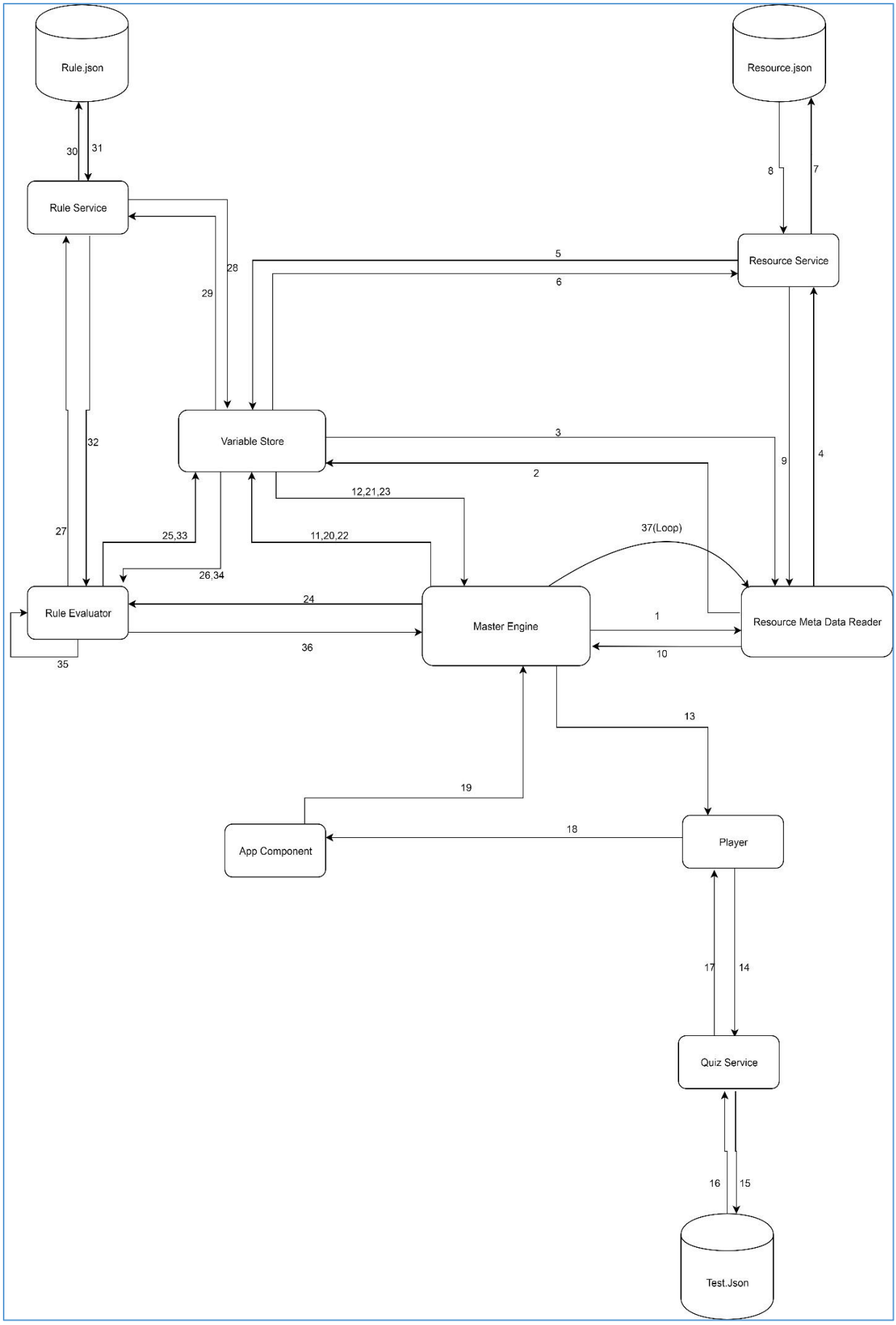
2.3.3 Flow Diagram for Question Type Resource

Steps	Function
1	Master Engine calling Resource Meta Data Reader
2,3	Getting Resource Id from Store
4	Calling Resource Service to get the Resource Object.
5,6	Getting Resource Path from Store.
7,8	Reads Resource.Json
9	Resource Service returns the Resource Object using Resource Id.
10	Returns the flow to Master Engine
11,12	Setting Variable Name
13	Flow goes to Player Component
14	Calling Question Service
15,16	Read Question.json
17	Returns Question object
18	Question is Displayed on App Component(Main Screen)
19	Click On Submit
20,21	Getting Variable Name from Store
22,23	Setting Variable Name in Store
24	Master engine calls the rule Evaluator for getting resource Id
25,26	Getting RuleId from store
27	Call Rule Service to get Rule Object using Rule Id.
28,29	Getting Rule Path from store
30,31	Rule Service reads Rule.json
32	Rule Service returns the Rule object to Rule Evaluator Service
33,34	Getting Variable value from Store
35	Evaluates the Rule if TrueAction or FalseAction is Evaluate Rule (Repeat from 25 to 34)
36	Returns to Master Engine
37	Repeat steps 1-36 if again the Resource Type Is Question.



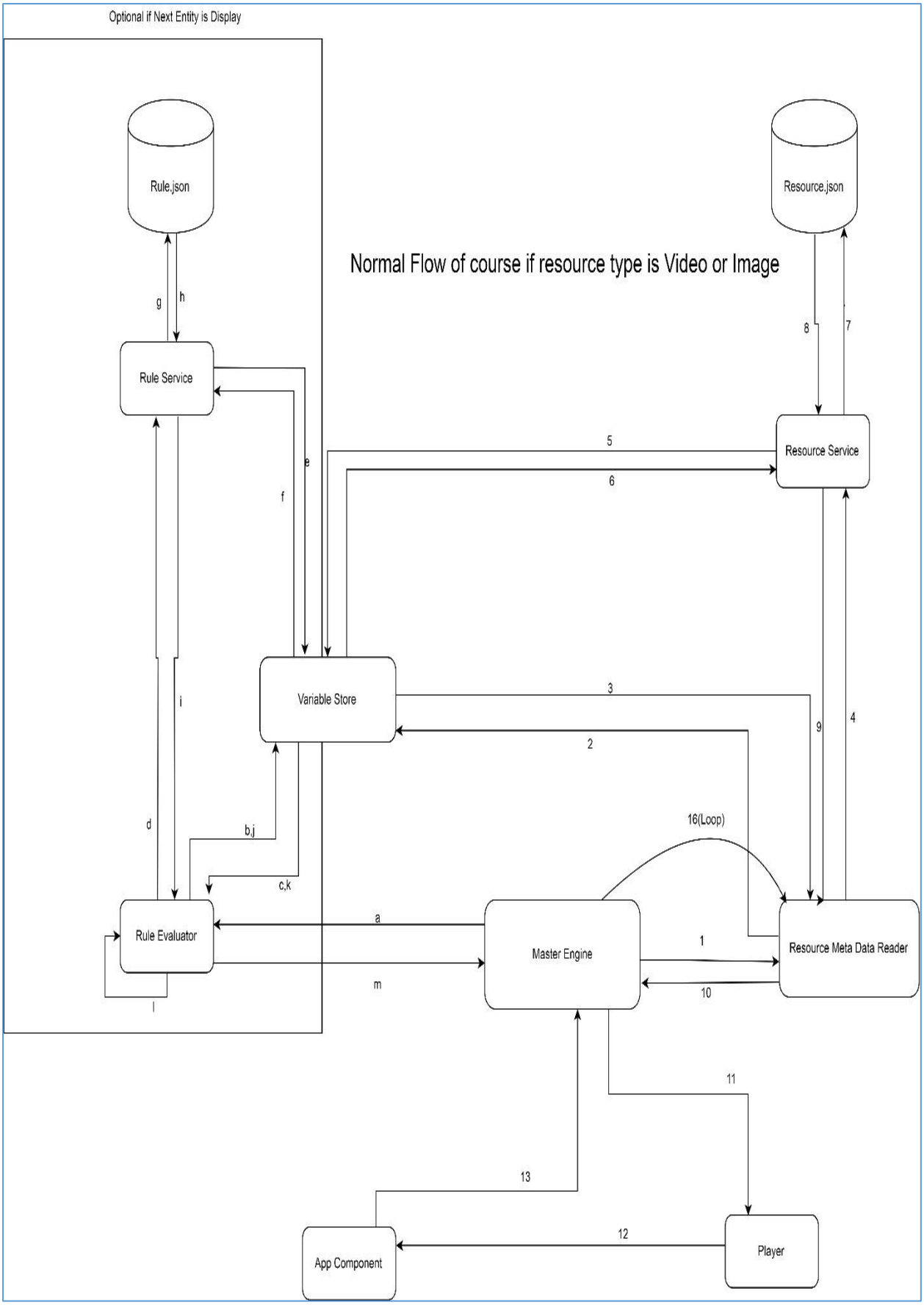
2.3.4 Flow Diagram for Quiz Type Resource

Steps	Function
1	Master Engine calling Resource Meta Data Reader
2,3	Getting Resource Id from Store
4	Calling Resource Service to get the Resource Object.
5,6	Getting Resource Path from Store.
7,8	Reads Resource.Json
9	Resource Service returns the Resource Object using Resource Id.
10	Returns the flow to Master Engine
11,12	Setting Variable Name
13	Flow goes to Player Component
14	Calling Question Service
15,16	Read Question.json
17	Returns Question object
18	Quiz is Displayed on App Component(Main Screen)
19	Click On Submit
20,21	Getting Variable Name from Store
22,23	Setting Variable Name in Store
a	Master engine calls the rule Evaluator for getting resource Id
b,c	Getting RuleId from store
d	Call Rule Service to get Rule Object using Rule Id.
e,f	Getting Rule Path from store
g,h	Rule Service reads Rule.json
i	Rule Service returns the Rule object to Rule Evaluator Service
j,k	Getting Variable value from Store
l	Evaluates the Rule if TrueAction or FalseAction is Evaluate Rule (Repeat from 25 to 34)
m	Returns to Master Engine
24	Loop



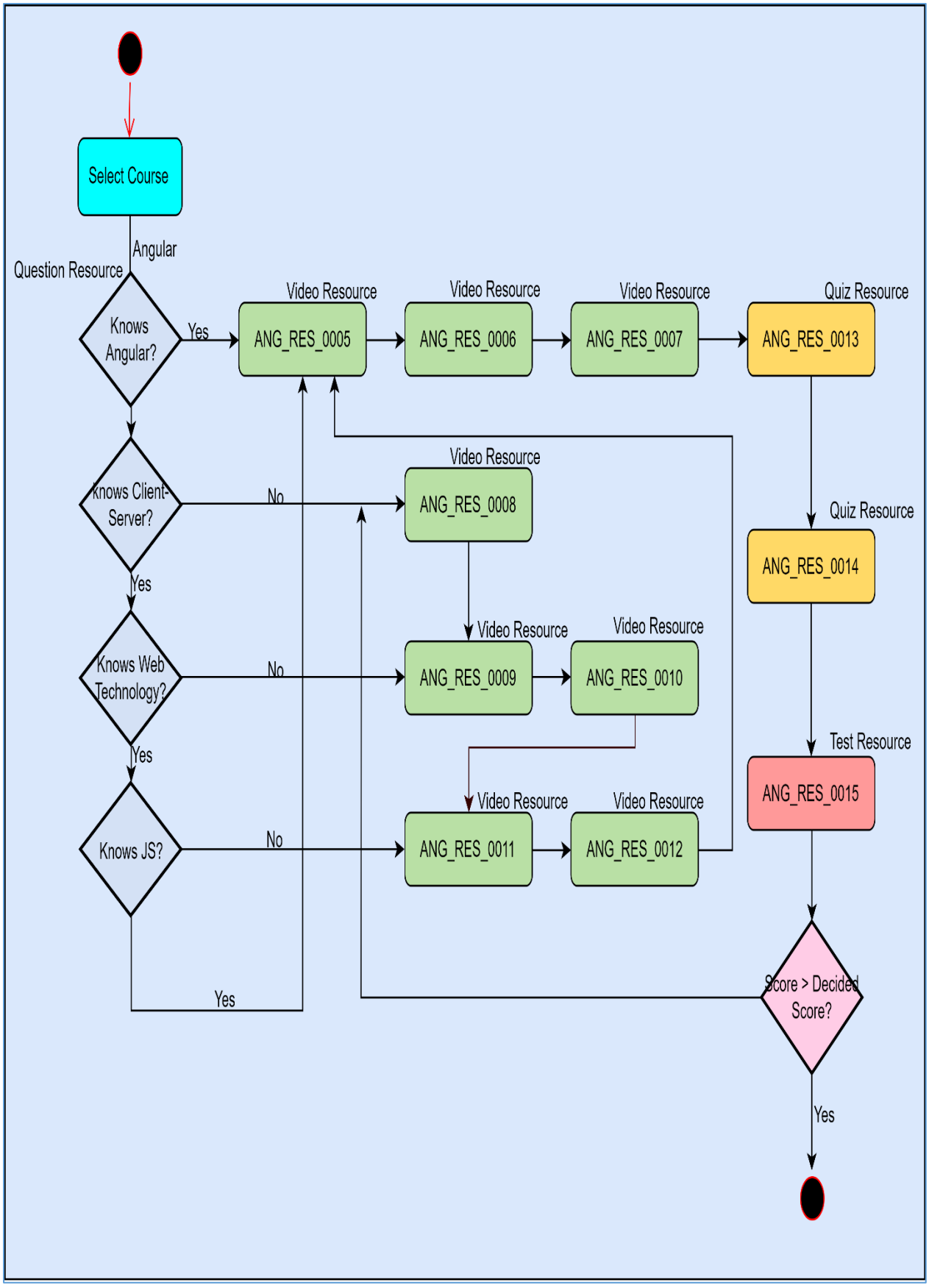
2.3.5 Flow Diagram for Test Type Resource

Steps	Function
1	Master Engine calling Resource Meta Data Reader
2,3	Getting Resource Id from Store
4	Calling Resource Service to get the Resource Object.
5,6	Getting Resource Path from Store.
7,8	Reads Resource.Json
9	Resource Service returns the Resource Object using Resource Id.
10	Returns the flow to Master Engine
11,12	Setting Variable Name
13	Flow goes to Player Component
14	Calling Quiz Service
15,16	Read Test.json
17	Returns Test object
18	Test is Displayed on App Component(Main Screen)
19	Click On Submit
20,21	Getting Variable Name from Store
22,23	Setting Variable Name in Store
24	Master engine calls the rule Evaluator for getting resource Id
25,26	Getting RuleId from store
27	Call Rule Service to get Rule Object using Rule Id.
28,29	Getting Rule Path from store
30,31	Rule Service reads Rule.json
32	Rule Service returns the Rule object to Rule Evaluator Service
33,34	Getting Variable value from Store
35	Evaluates the Rule if TrueAction or FalseAction is Evaluate Rule (Repeat from 25 to 34)
36	Returns to Master Engine
37	Loop



2.3.6 Flow Diagram for Video and Image Type Resource

Steps	Function
1	Master Engine calling Resource Meta Data Reader
2,3	Getting Resource Id from Store
4	Calling Resource Service to get the Resource Object.
5,6	Getting Resource Path from Store.
7,8	Reads Resource.Json
9	Resource Service returns the Resource Object using Resource Id.
10	Returns the flow to Master Engine
11	Flow goes to Player Component
12	Resource (Video or Image) is displayed on the App component (Main Screen)
13	Click event occurs (Next / Previous)
a	Master engine calls the rule Evaluator for getting resource Id
b,c	Getting RuleId from store
d	Call Rule Service to get Rule Object using Rule Id.
e,f	Getting Rule Path from store
g,h	Rule Service reads Rule.json
i	Rule Service returns the Rule object to Rule Evaluator Service
j,k	Getting Variable value from Store
l	Evaluates the Rule if TrueAction or FalseAction is Evaluate Rule (Repeat from 25 to 34)
m	Returns to Master Engine
14	Loop



2.3.7 Working of Angular Course

3.PROJECT FLOW

3.1 THE CONCEPT

The concept of the system is to be able to accommodate all the courses under one learning platform that should follow some proper flow and yet remain dynamic to the content and the structure of the course. In order to achieve this flow, multiple components have been introduced to achieve the following objectives:

1.Modularity

2.Dynamicity

3.Adaptive

4.Customized

3.2 COMPONENT, SERVICES AND STORE

3.2.1 MASTER ENGINE SERVICE

It is the main service in this application. It communicates with all the other service and component and transfer data across them. It makes call to one component, service and fetches the required data and pass the required to another component, service as and when required.

3.2.2 RULE EVALUATOR SERVICE

It is responsible for the evaluation of the rule that are given in the json file and returns resource id to master engine. The Rules are evaluated for displaying what resource to be played based on the input from user. Rule Evaluator checks for the equality, greater than, less than, greater than or equal to, less than or equal to, AND, OR functions.

3.2.3 RULE SCHEMA READER SERVICE

It directly communicate with Rule service to get the rule object and returns to Rule Evaluator Service.

3.2.4 RESOURCE META DATA READER SERVICE

It deals with Resource.json and returns the resource location and resource type of the resource according to resource id passed to it as a parameter.

3.2.5 VARIABLE STORE

It is used store the variable name associated to the resource and its corresponding value after getting it from the user as variable value in the store.

3.2.6 COURSE-LIST COMPONENT

It is first page after running the application that fetch the course name from Variable Store and show it to the user.

3.2.7 PLAYER COMPONENT

It is used to display all the type of resources that includes image, video, prerequisite questions, quizzes and assessments. These resources are only played only after the rules are evaluated based on the responses received from the user.

3.2.8 RULE SERVICE

This file makes http call to load rule.json in application.

3.2.9 RESOURCE SERVICE

This file makes http call to load resource.json in application.

3.2.10 COURSE SERVICE

This file make http call to load course.json in application.

4. IMPLEMENTATION

4.1 COURSE-LIST

The first thing that is displayed on the UI is the list of courses. Therefore, first the format of course.json is made, so that what all courses are to be displayed on the screen can be read from the file.
Based on the course selected by the user, the variable name ‘Course’ is updated with the value as name of the Course.

4.2 MASTER ENGINE

The Master engine is implemented as an Angular service that can be used by all the components to call the various other component as and when required.

4.3 VARIABLE STORE

The variable store is used to store the value of variable dynamically. The variable names are also specific to that particular resource.

4.4 RULE EVALUATOR

Rule evaluator evaluates the rule using the operator functions such as checkEquality(), checkGreaterThan(), checkLessThan(), etc. based on the return type resources are displayed or next rule is evaluated.

4.5 RULE SCHEMA READER

Rule schema Reader reads the Rules.json file based on the response provided to Master Engine. Rule object fetched is then forwarded to Rule Evaluator.

4.6 RESOURCE META DATA READER

Resource Meta Data Reader fetches resource object using resource id from Resource.json. It returns the resource type and location to Master Engine.

4.7 QUESTION.JSON

This json contains different question types separately. Type of questions included are single choice, multiple choice and interaction.

4.8 COURSE.JSON

It contains list of all the courses and variable name identifiers for each course. It is read at the starting of the application when user selects the course.

4.9 TEST.JSON

It contain different kind of question such as multiple choice, single choice.

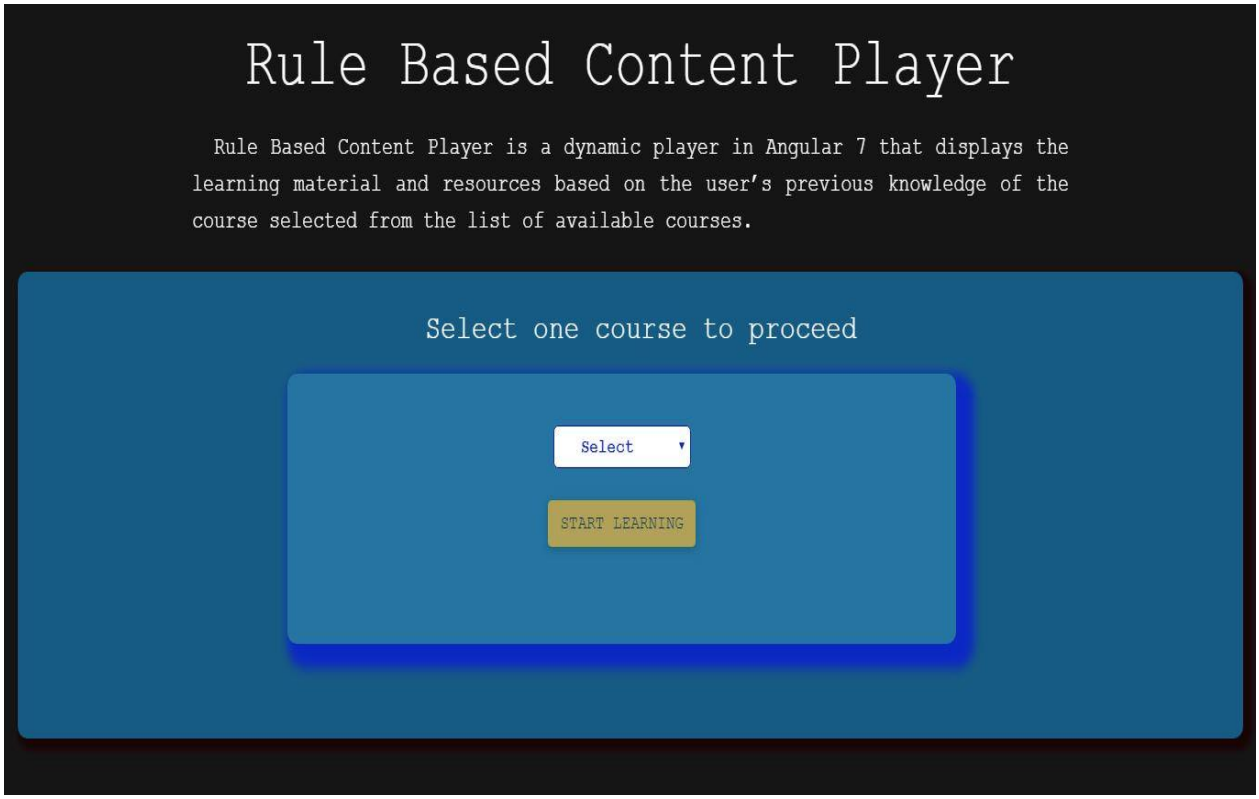
4.10 RESOURCE.JSON

It contains the format in which different type of resources for a particular course will be stored with required key-value pair for each type of resource.

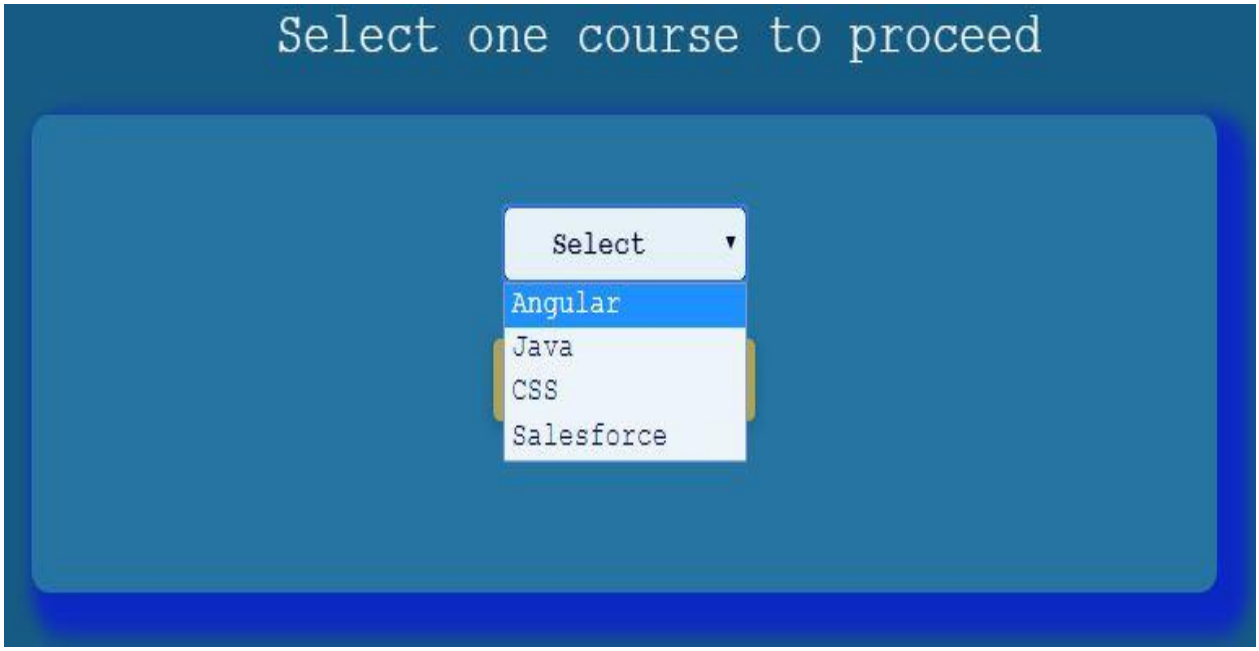
4.11 RULES.JSON

It contains the format in which all rules for a particular course will be stored with required key-value pair.

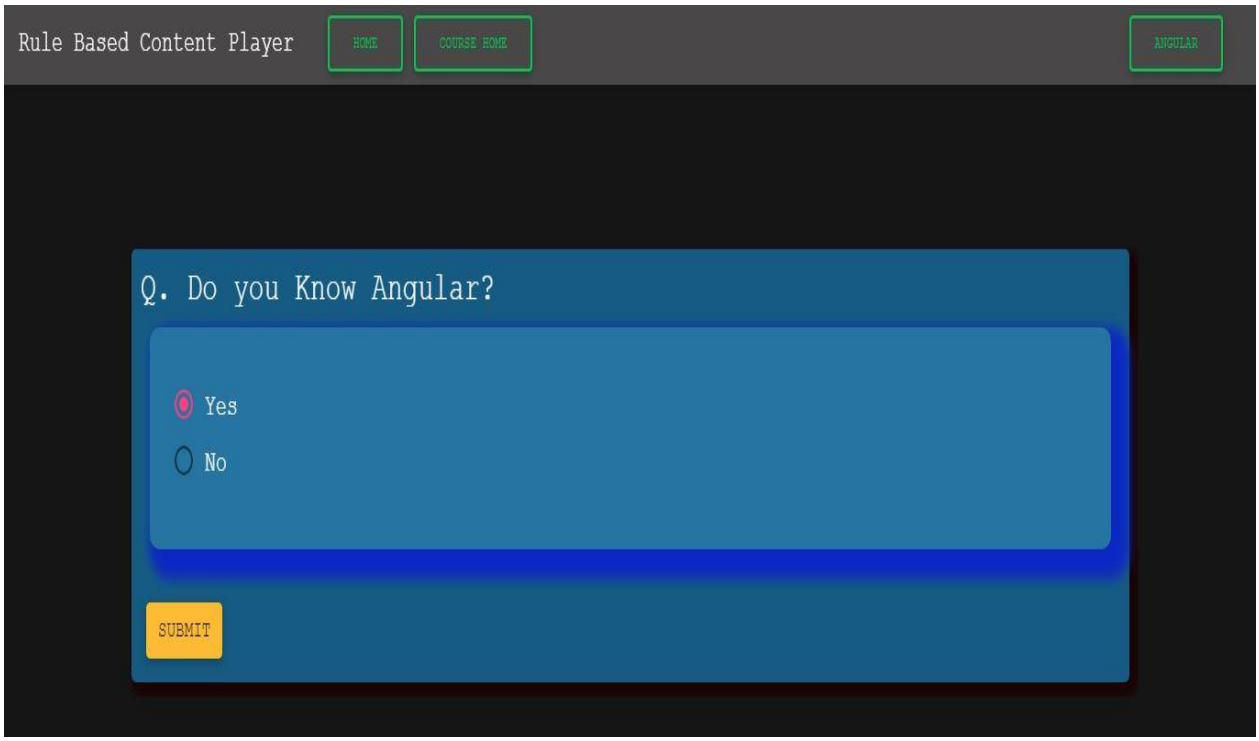
4.12 SCREENSHOTS



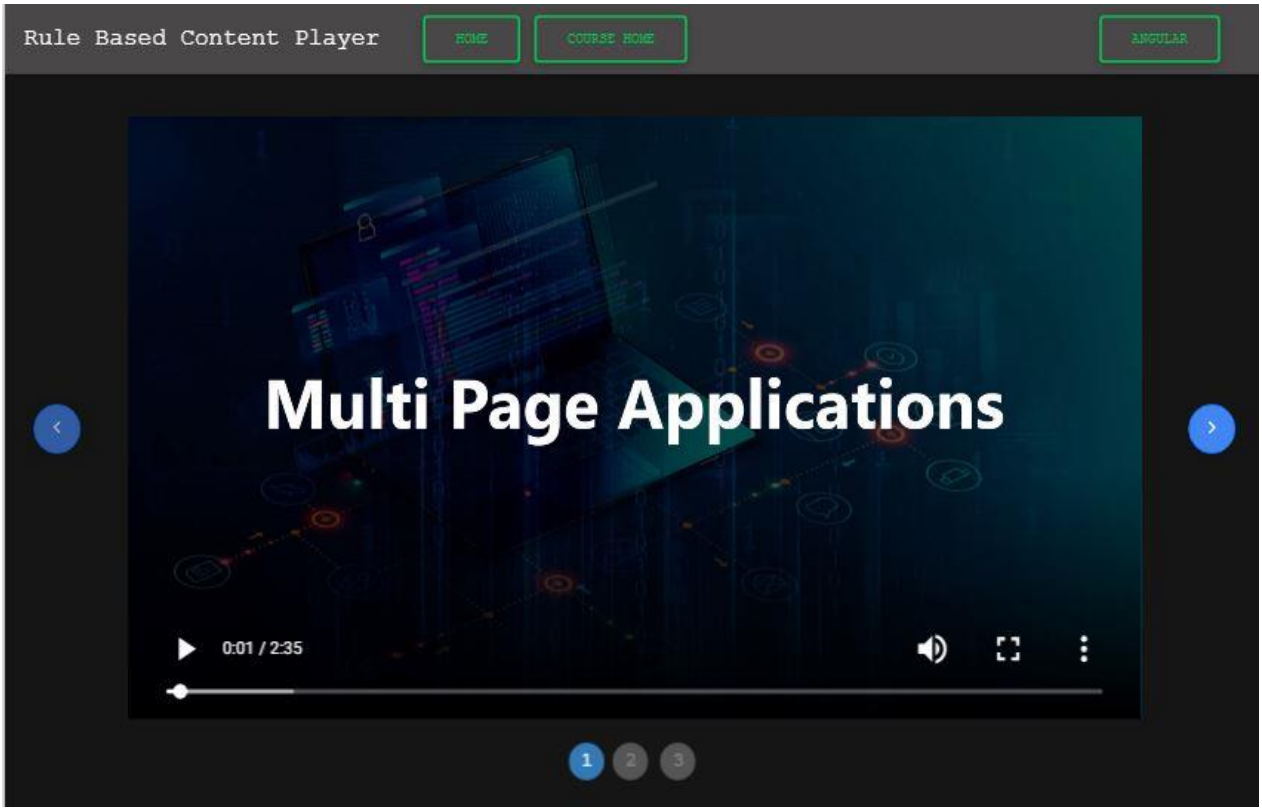
4.12.1 Home Page



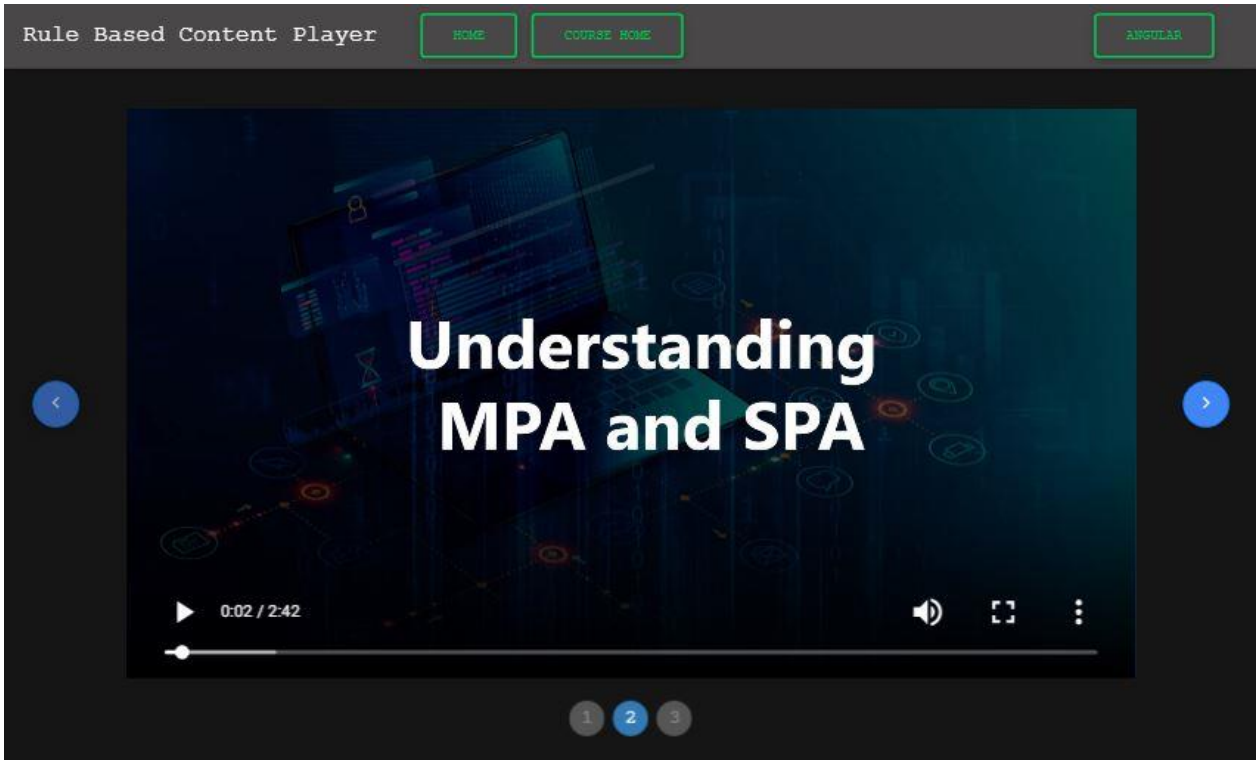
4.12.2 Select course from list of courses.



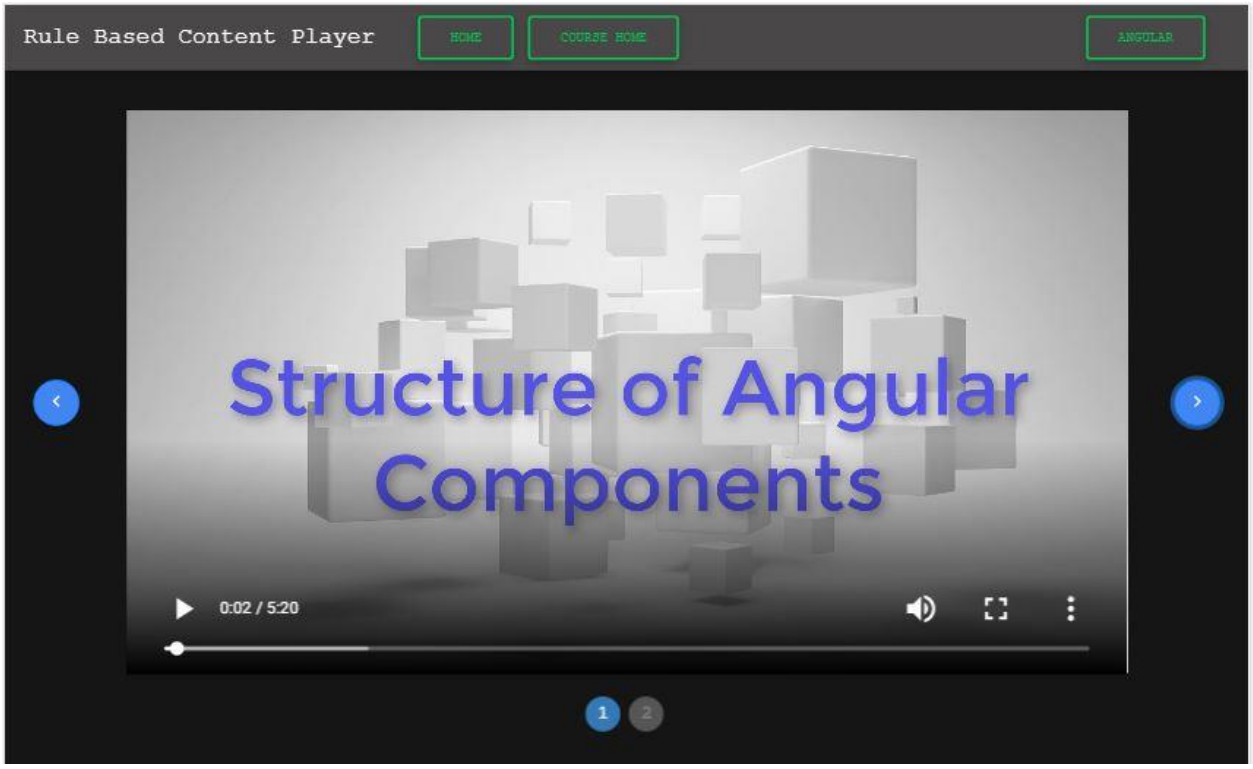
4.12.3 The Question resource for prerequisite questions.



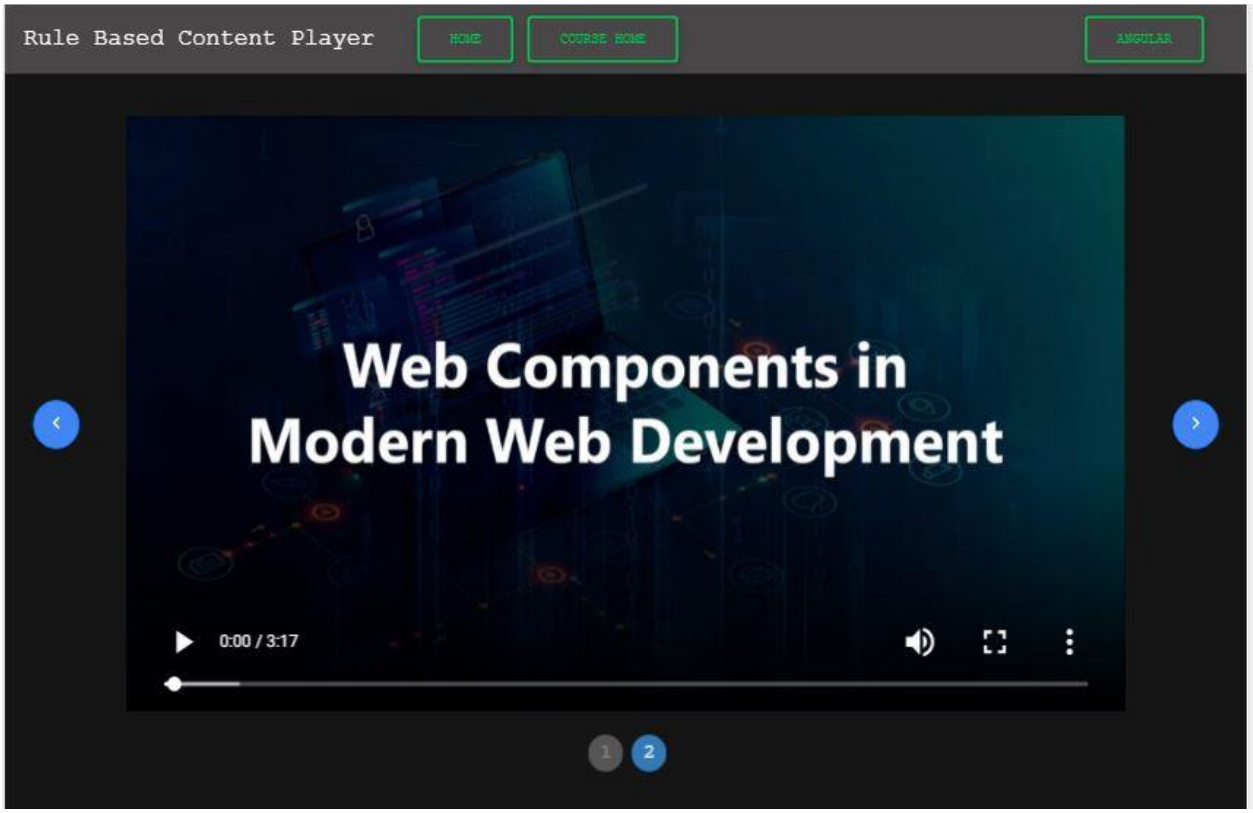
4.12.4 The first video resource for the course if the user knows Angular



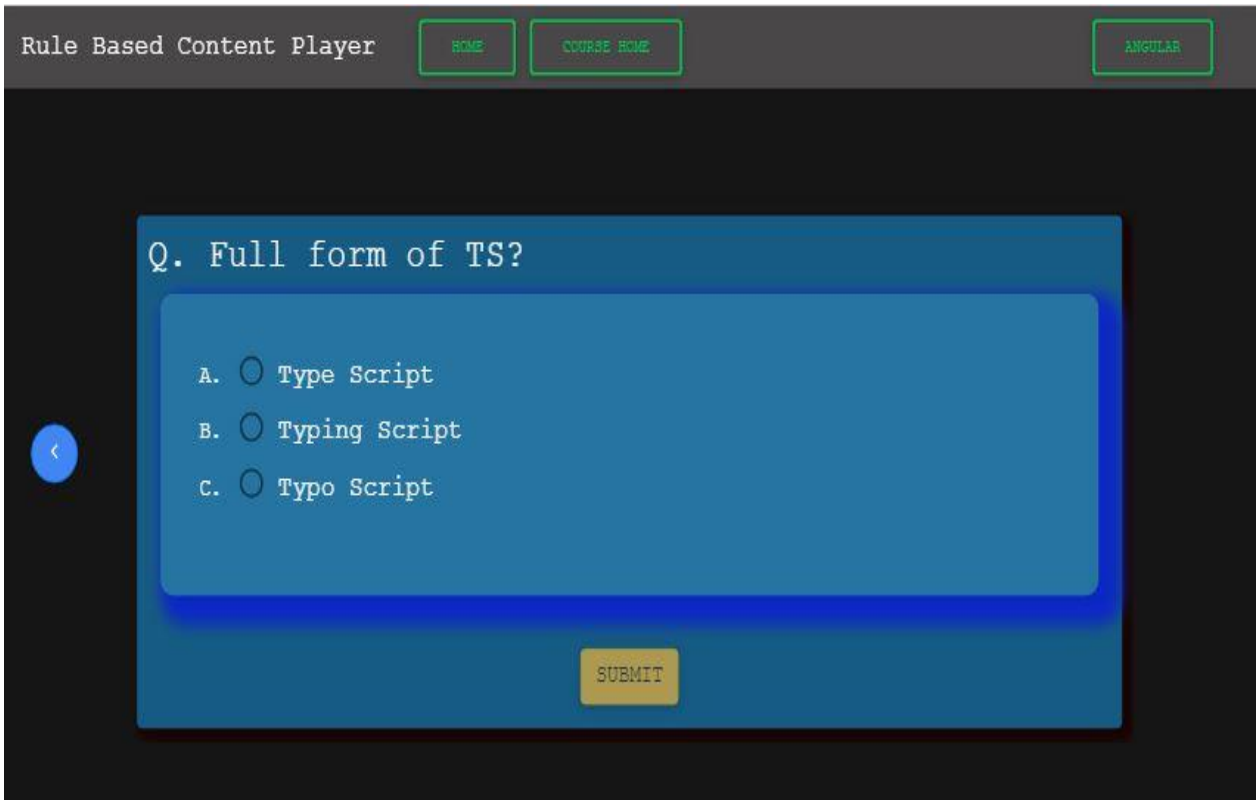
4.12.5 The second video resource for the course if the user knows Angular



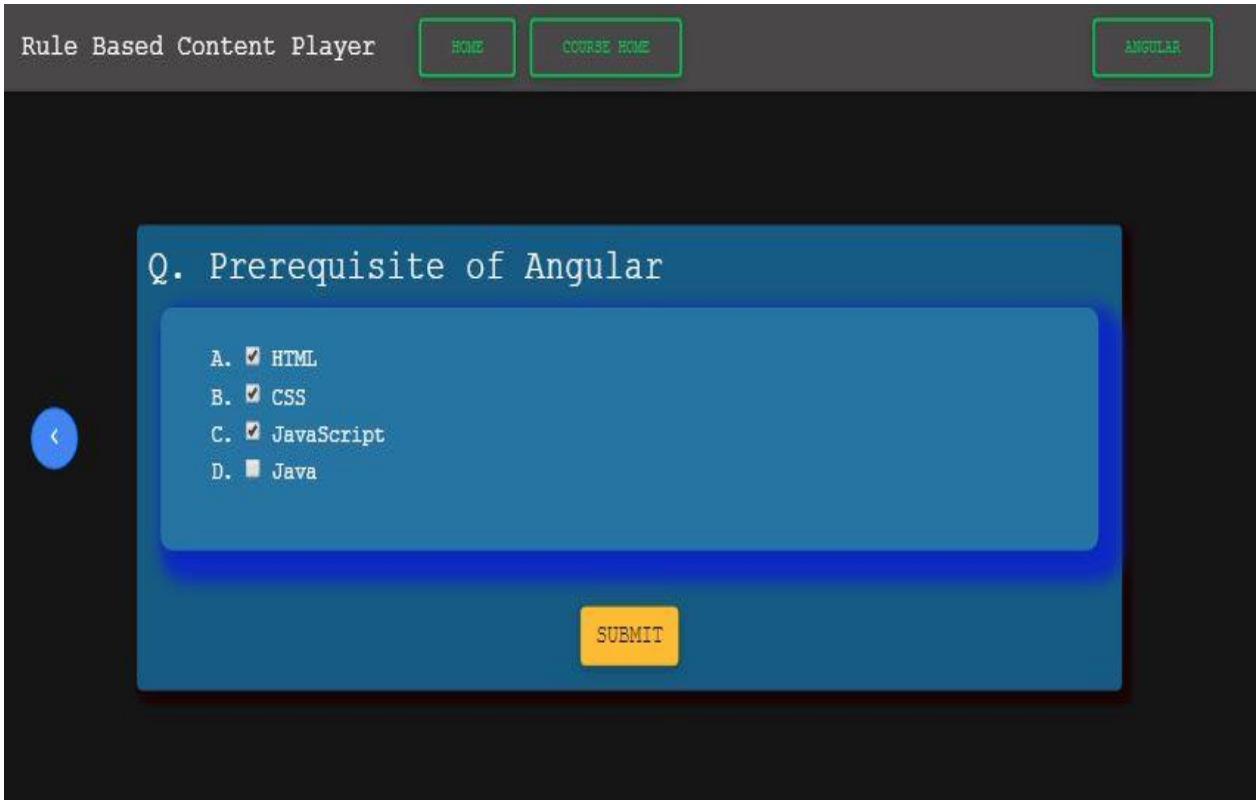
4.12.6 The third video resource for the course if the user knows Angular



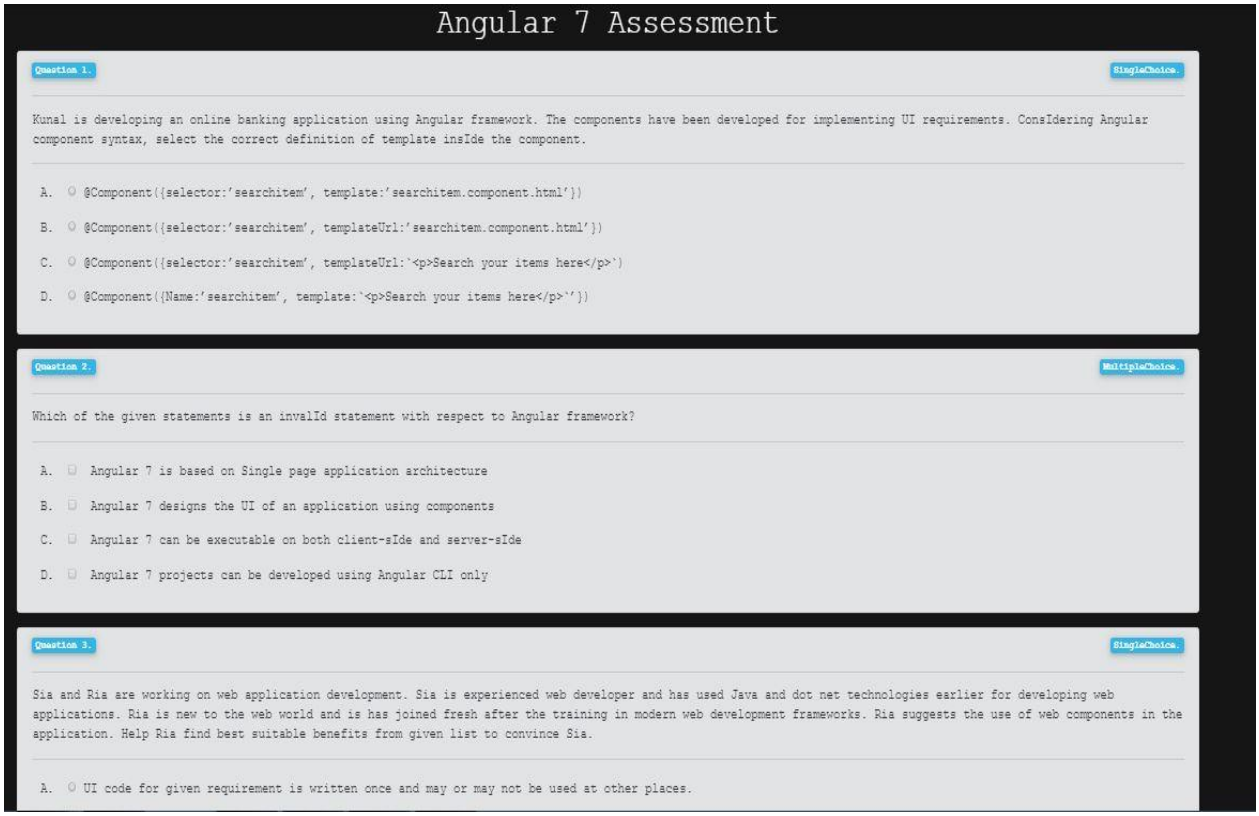
4.12.7 The video resource displayed if the user doesn't know Angular.



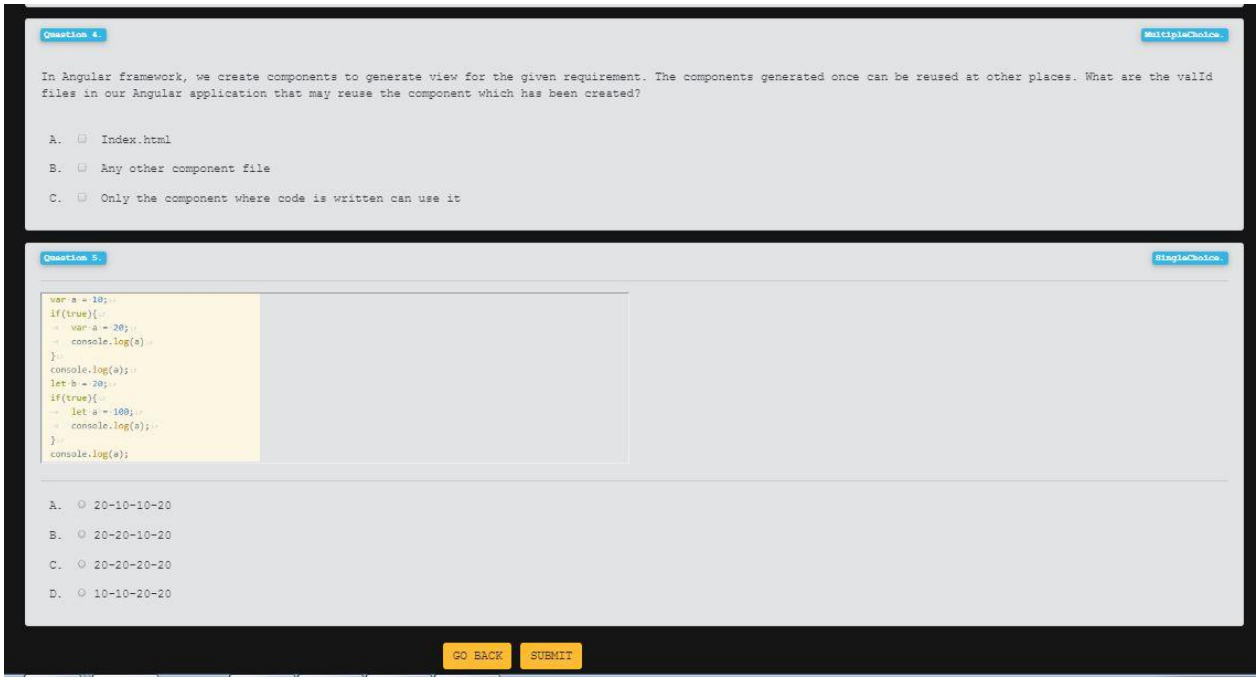
4.12.8 Quiz type resource (Single choice resource).



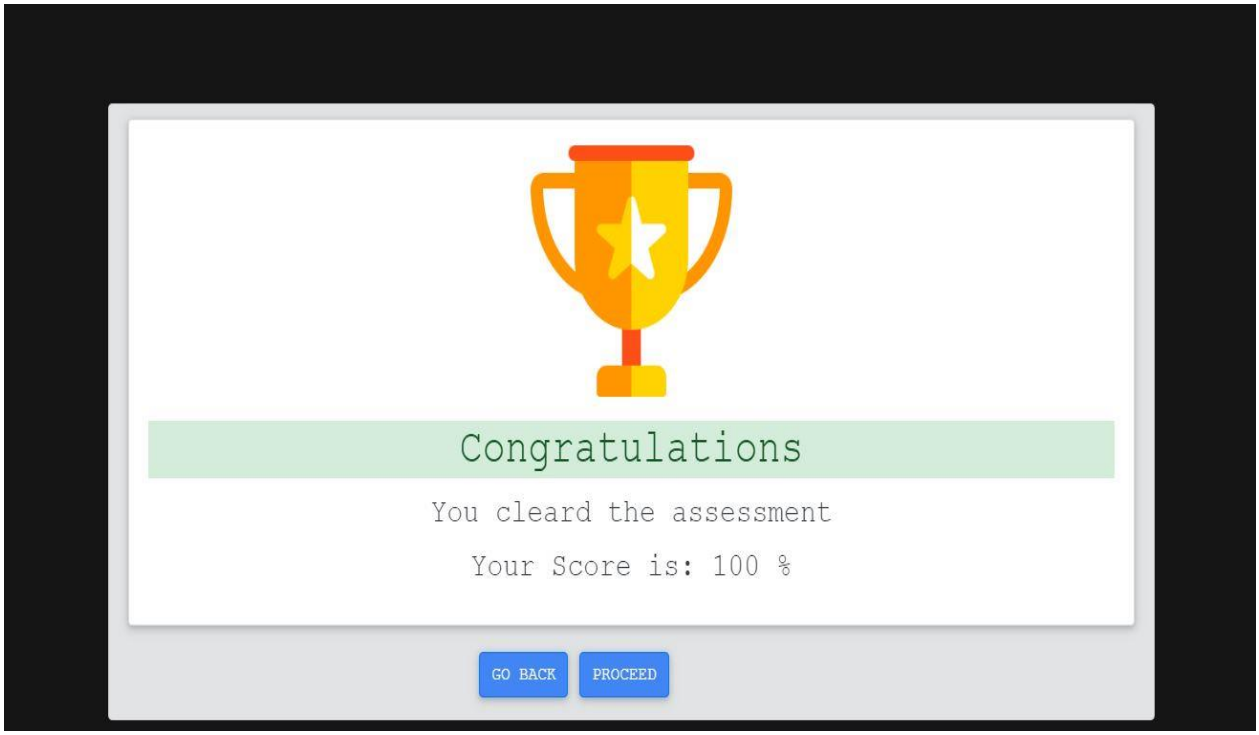
4.12.9 Quiz type resource (Multiple choice resource).



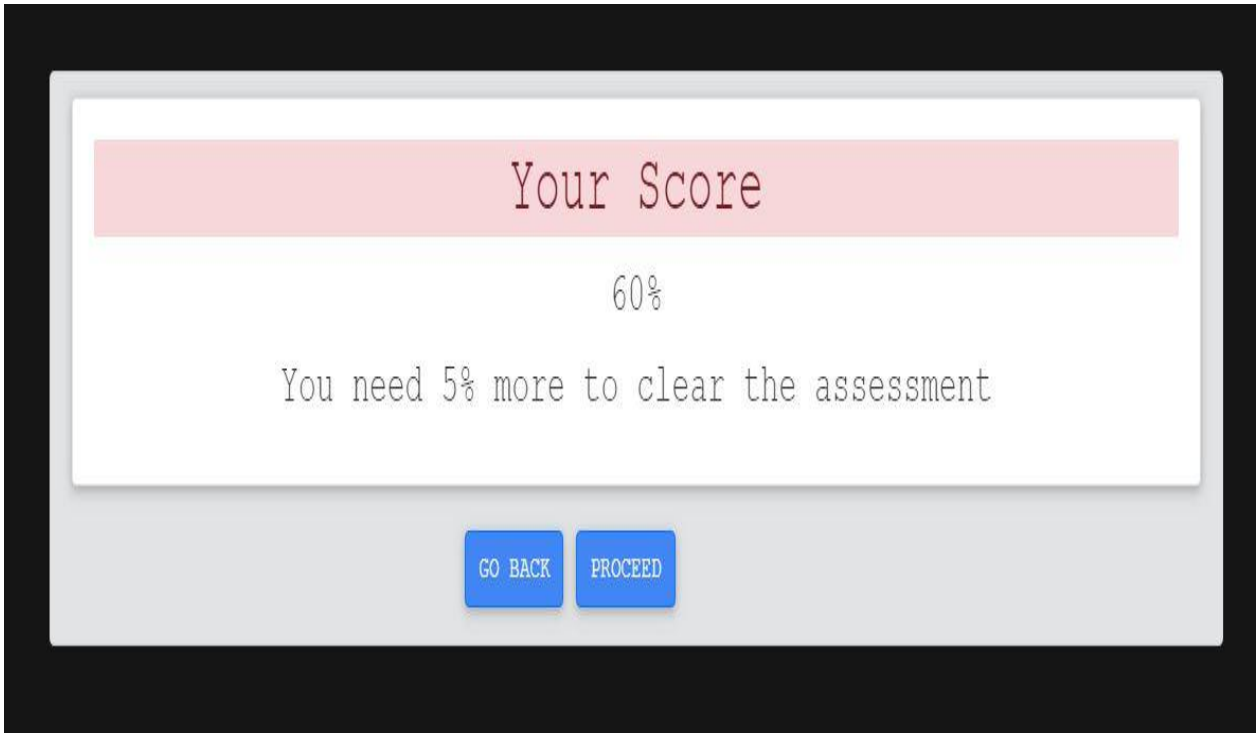
4.12.10 Test type Resource (part 1).



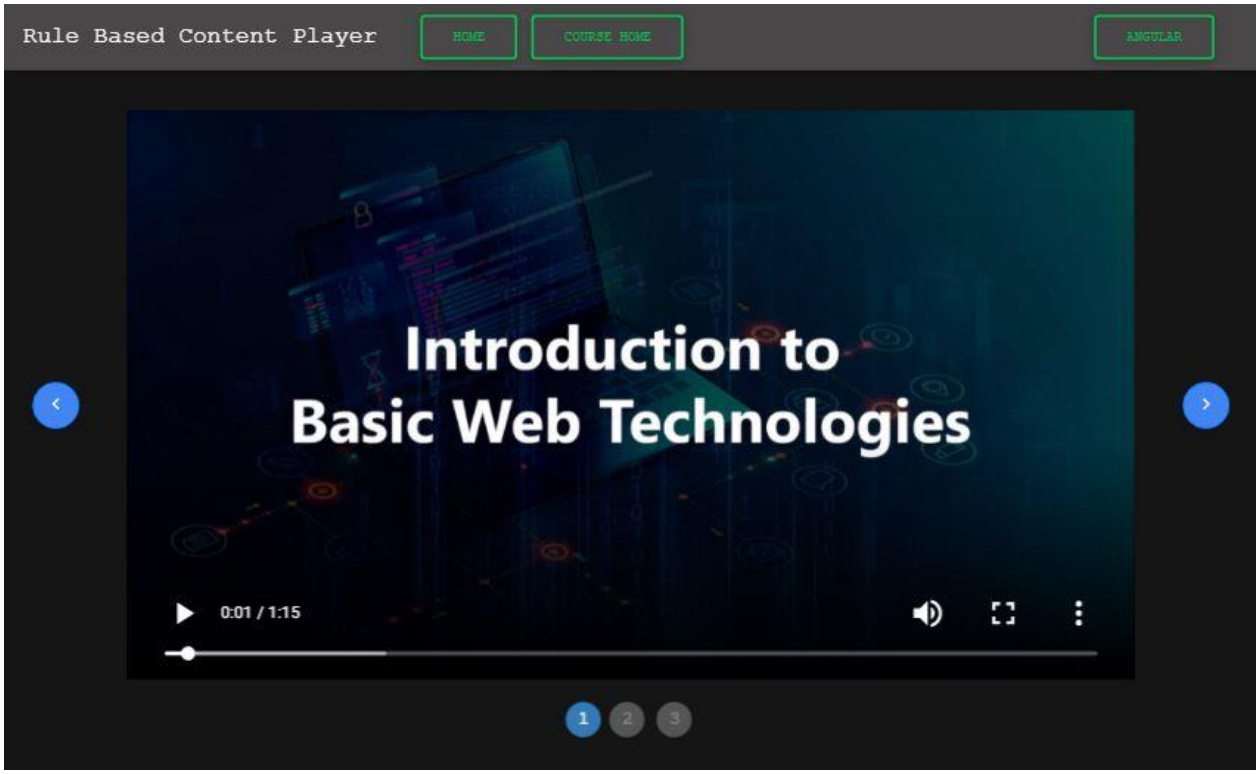
4.12.11 Test type Resource (part 2).



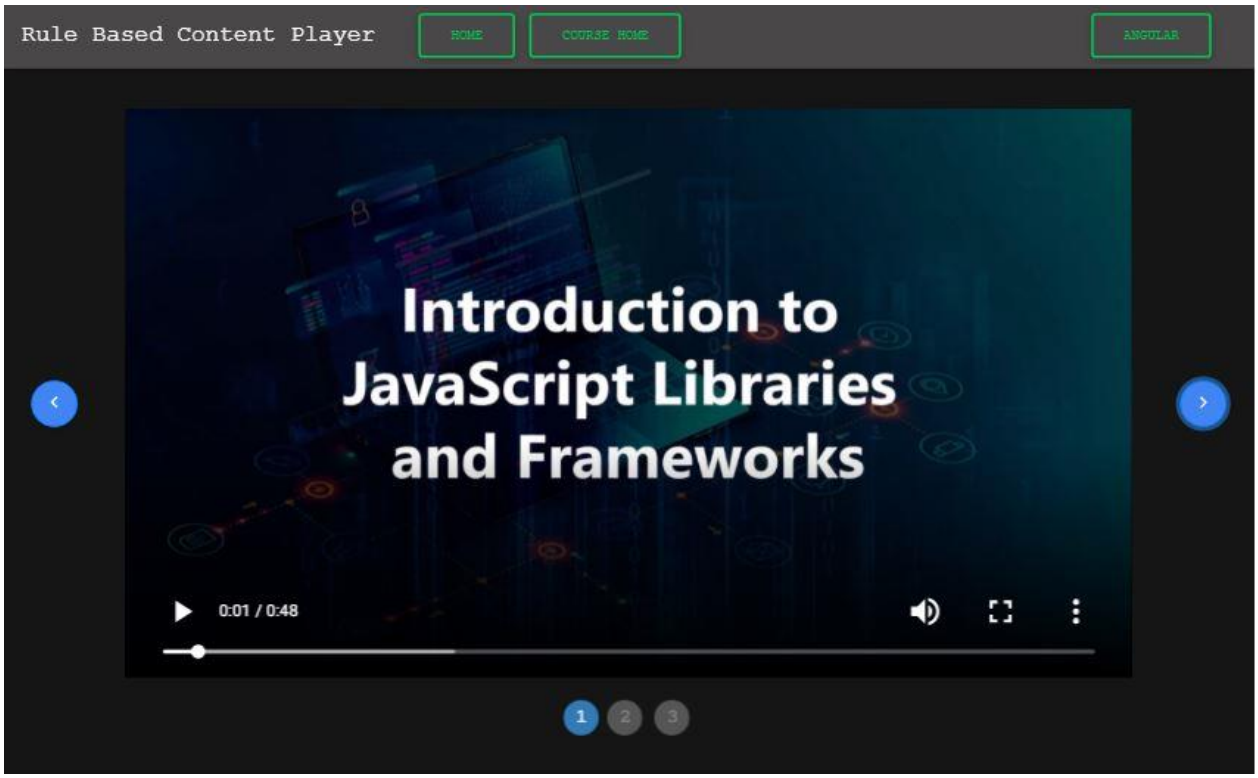
4.12.12 The Score Page (if the score is greater than decided score)



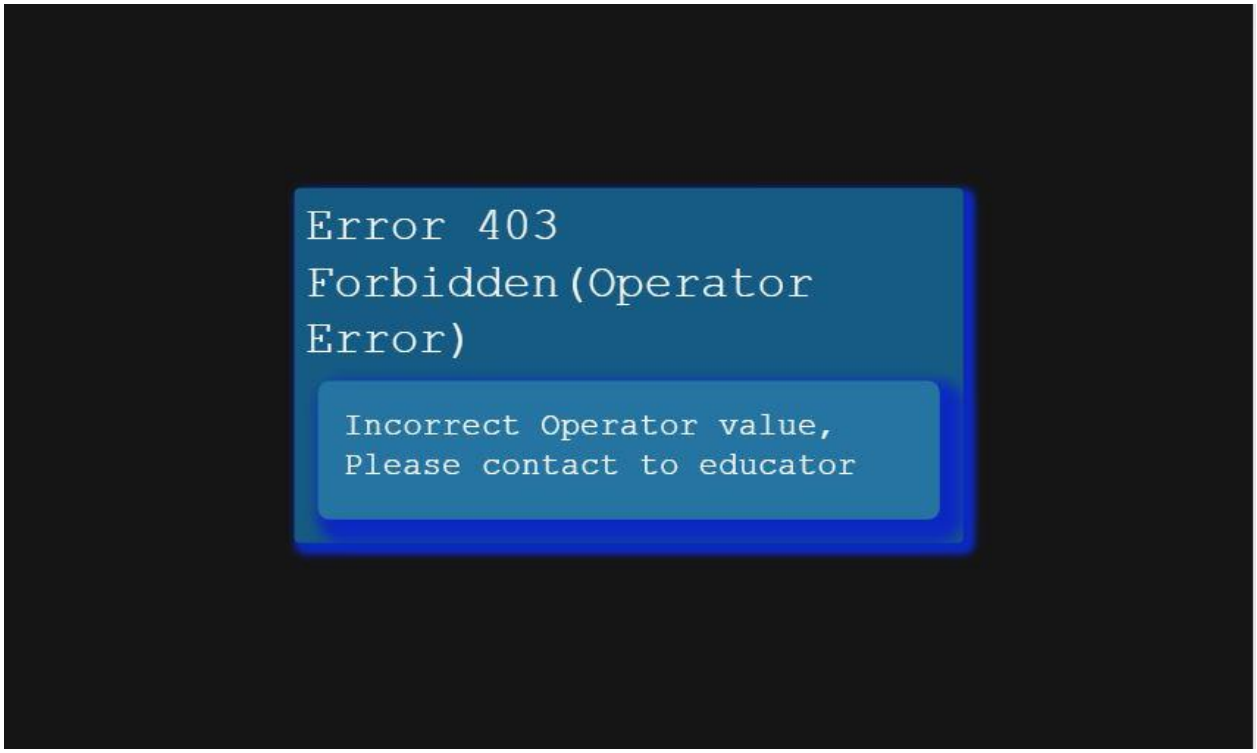
4.12.13 The Score Page (if the score is less than decided score)



4.12.14 The Video resource 1 for the score less than decided score



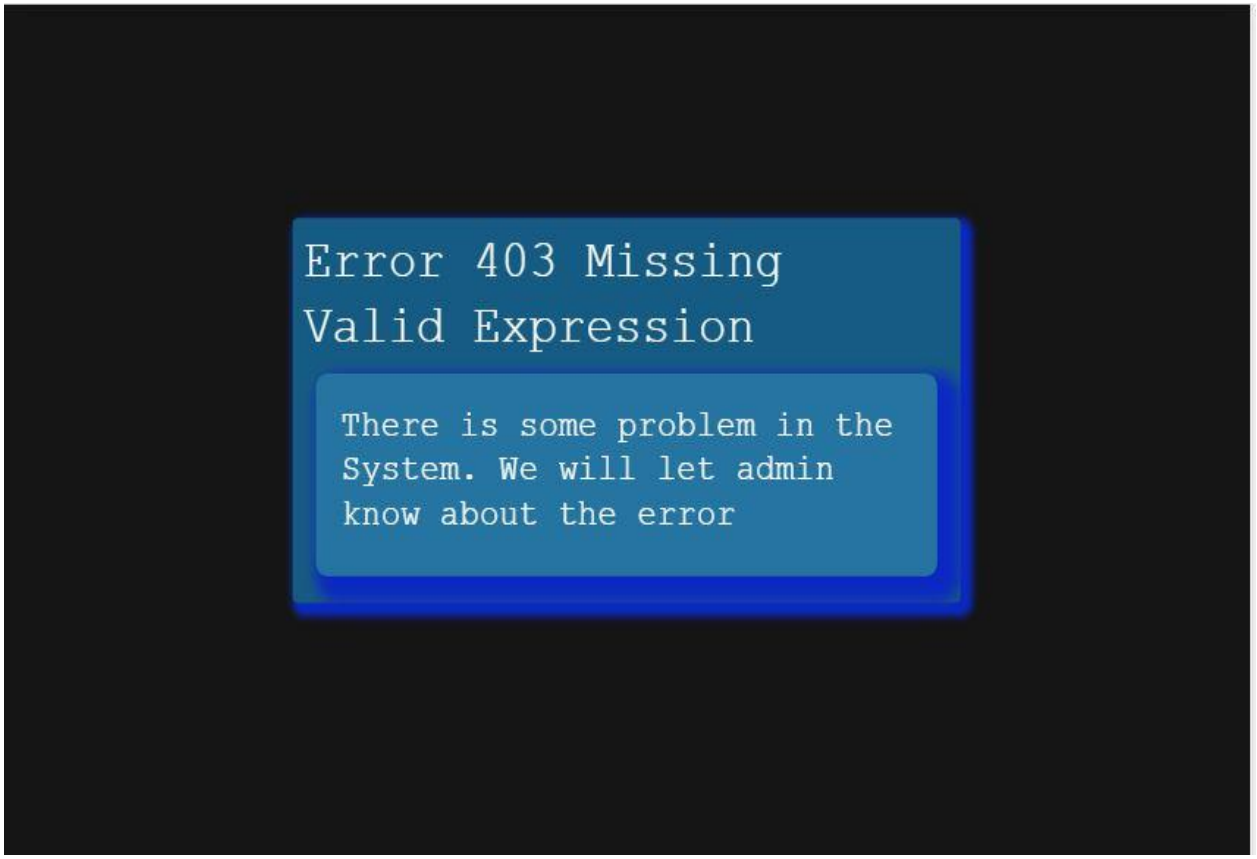
4.12.15 The Video resource 2 for the score less than decided score



4.12.16 The error component if there is a wrong operator value in json.

4.12.17 The error component if there is an Invalid resource id in json file.

4.12.18 The error component if there is an Invalid rule in json.



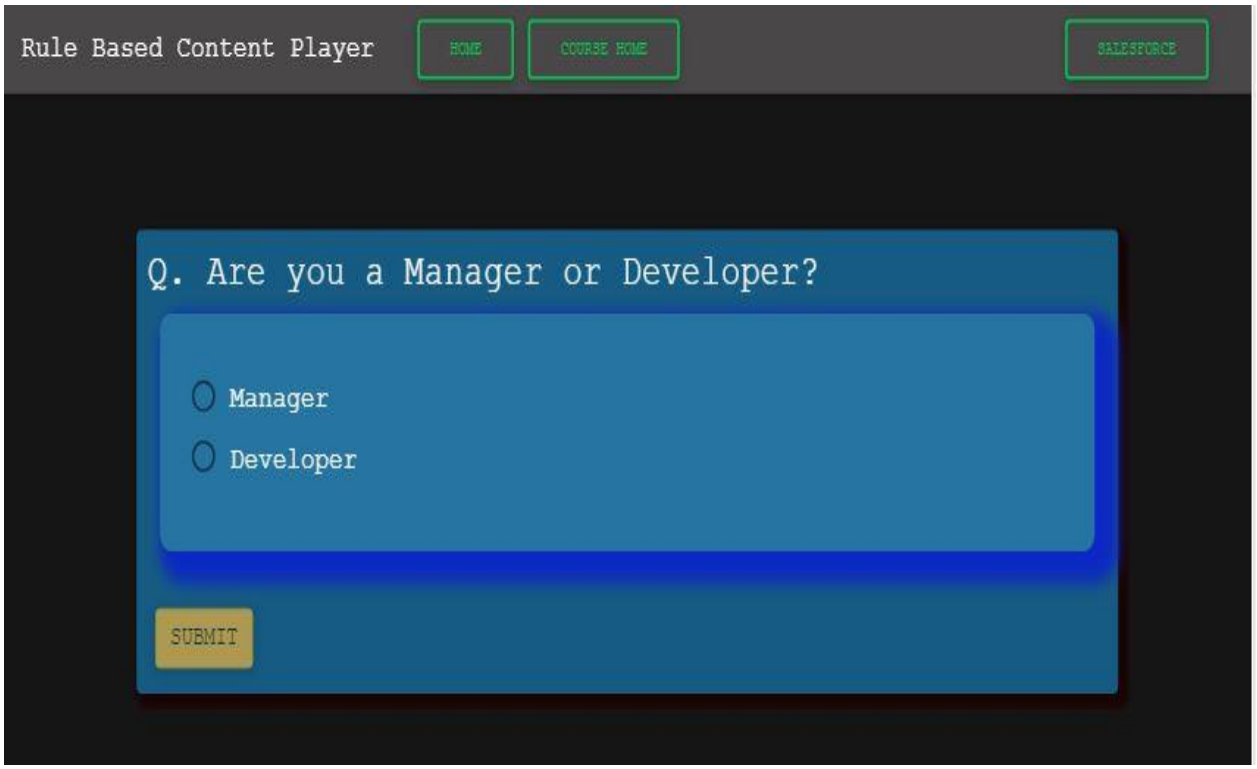
4.12.19 The error component if there is an invalid expression in the json.

4.12.20 The error component if there is a problem in main component.

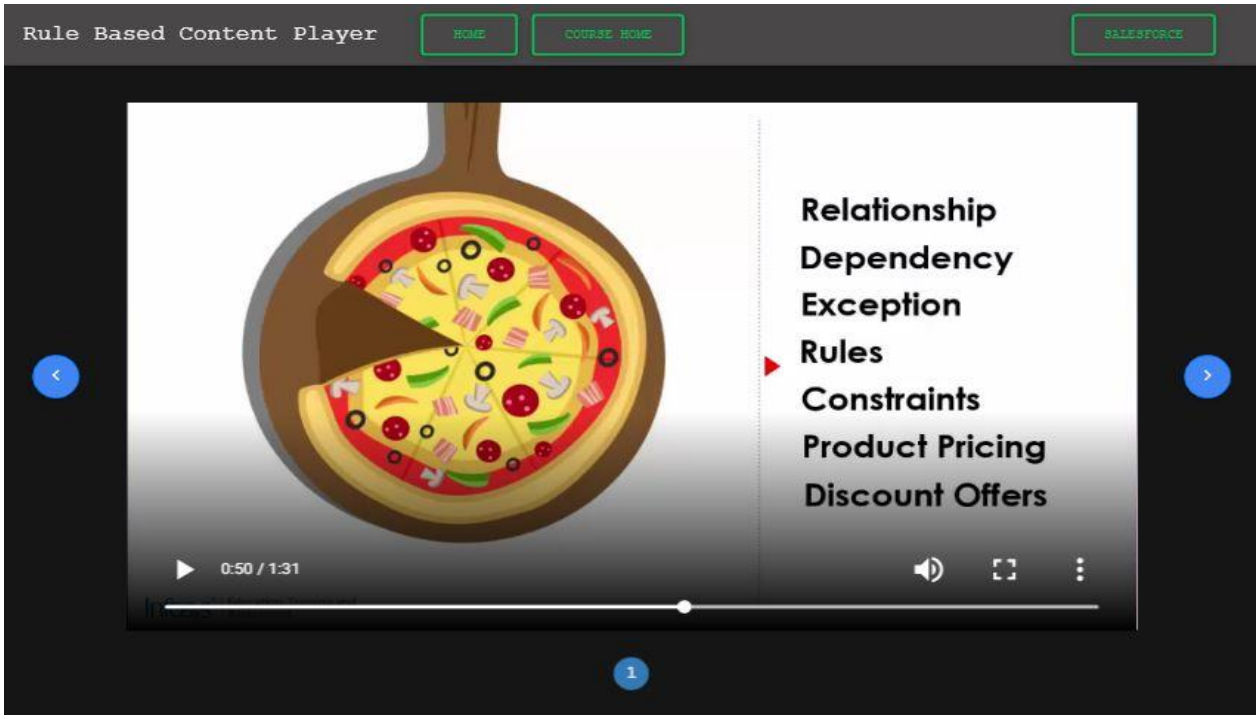
5.TESTING

There are different types of testing carried out for the system to be able to work properly in different scenarios.

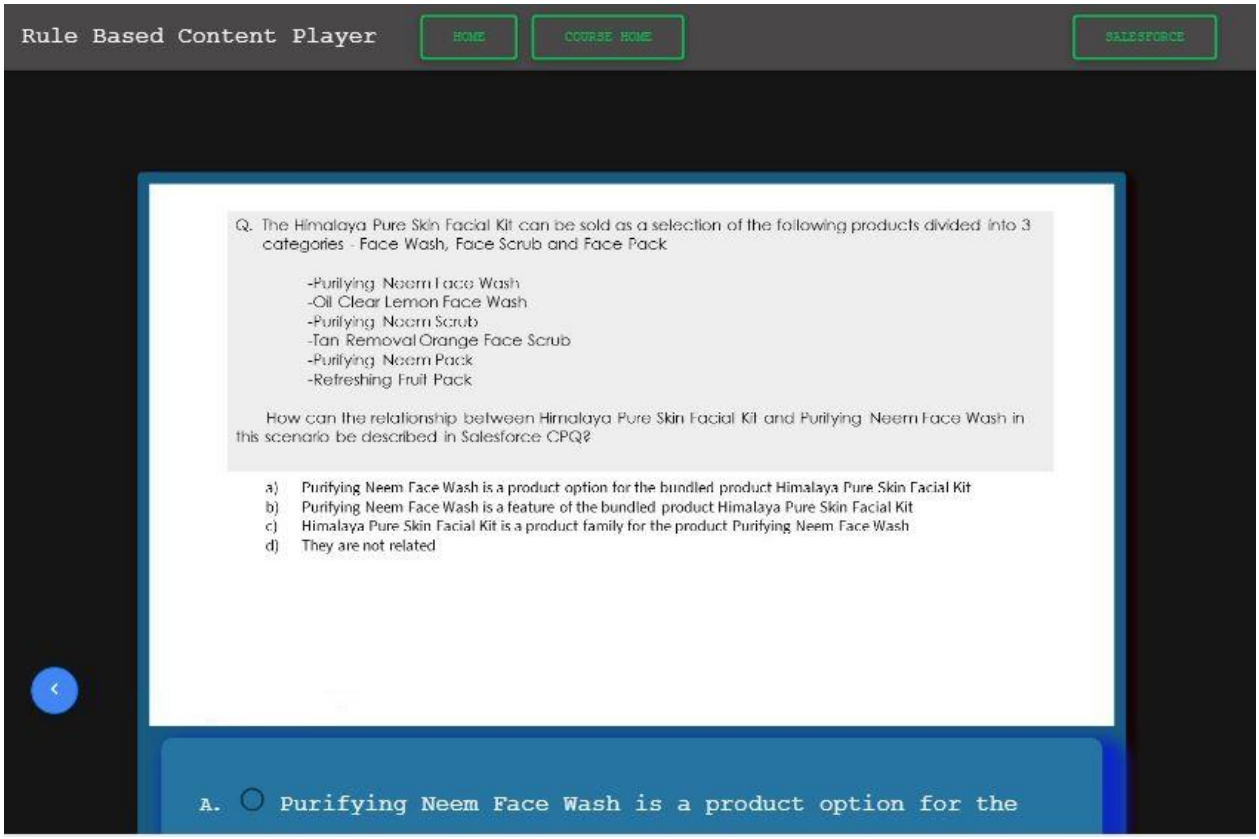
1. Unit Test: Checking each component for any problem.
2. Integration Test: After integration of each component, the system needs to be tested. Our system has 2 component and integration testing after adding each component was done.
3. Regression Test: Our system was tested for different courses by providing new Json files for Rule, Resource and Question component.



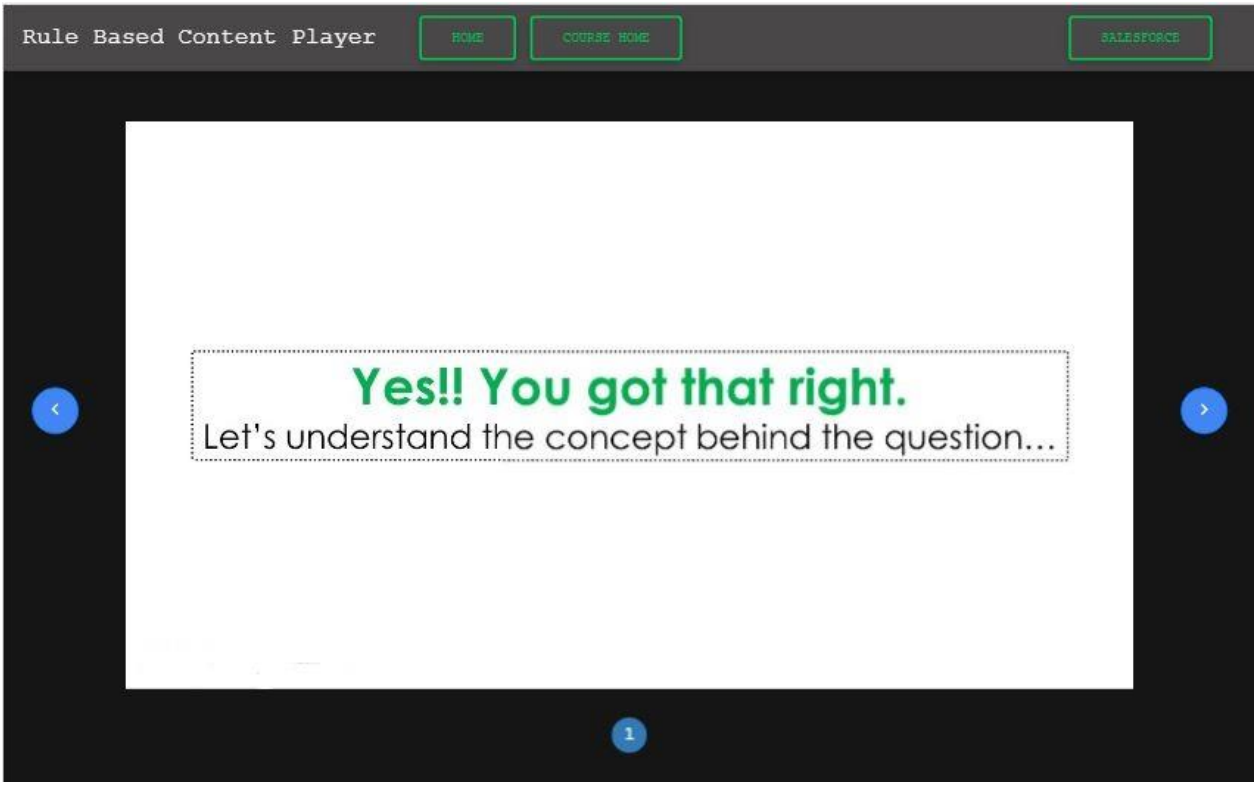
5.1.1 Prerequisite question for new Course (dynamically added).



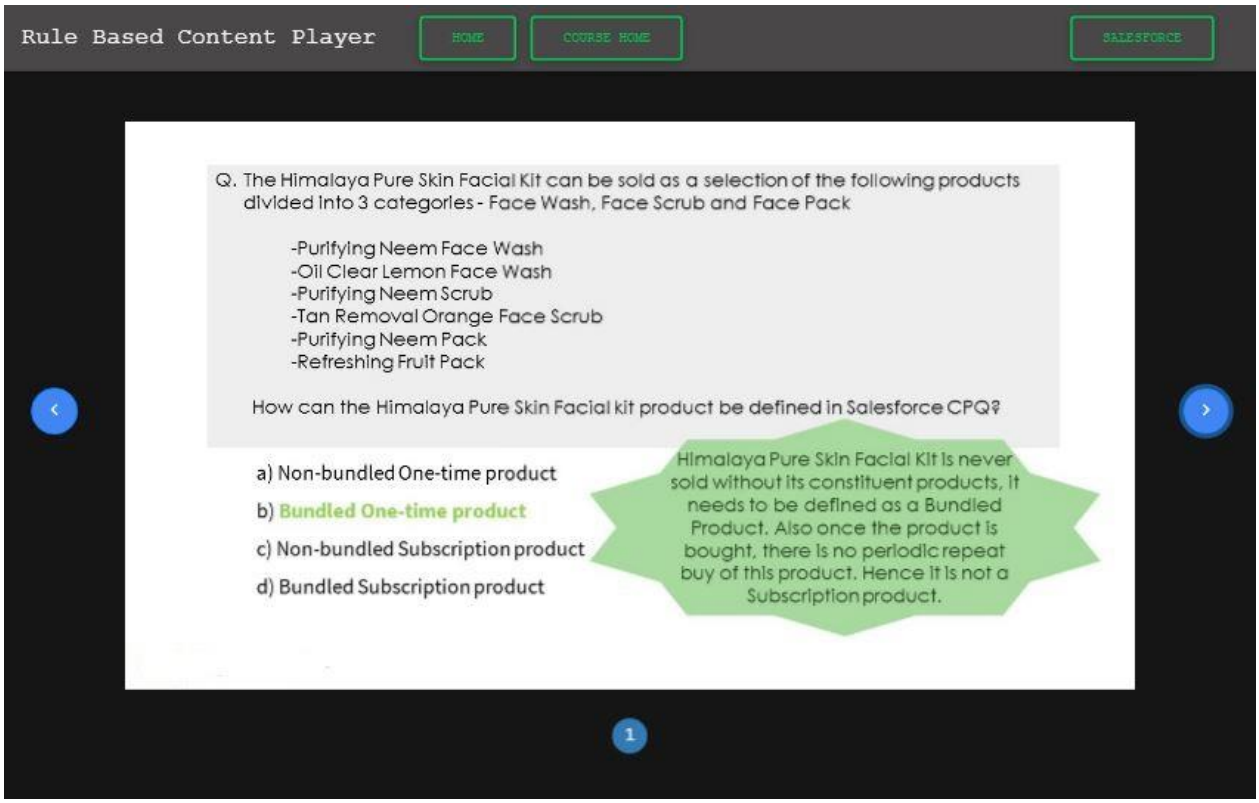
5.1.2 Introduction page for new course (added dynamically)



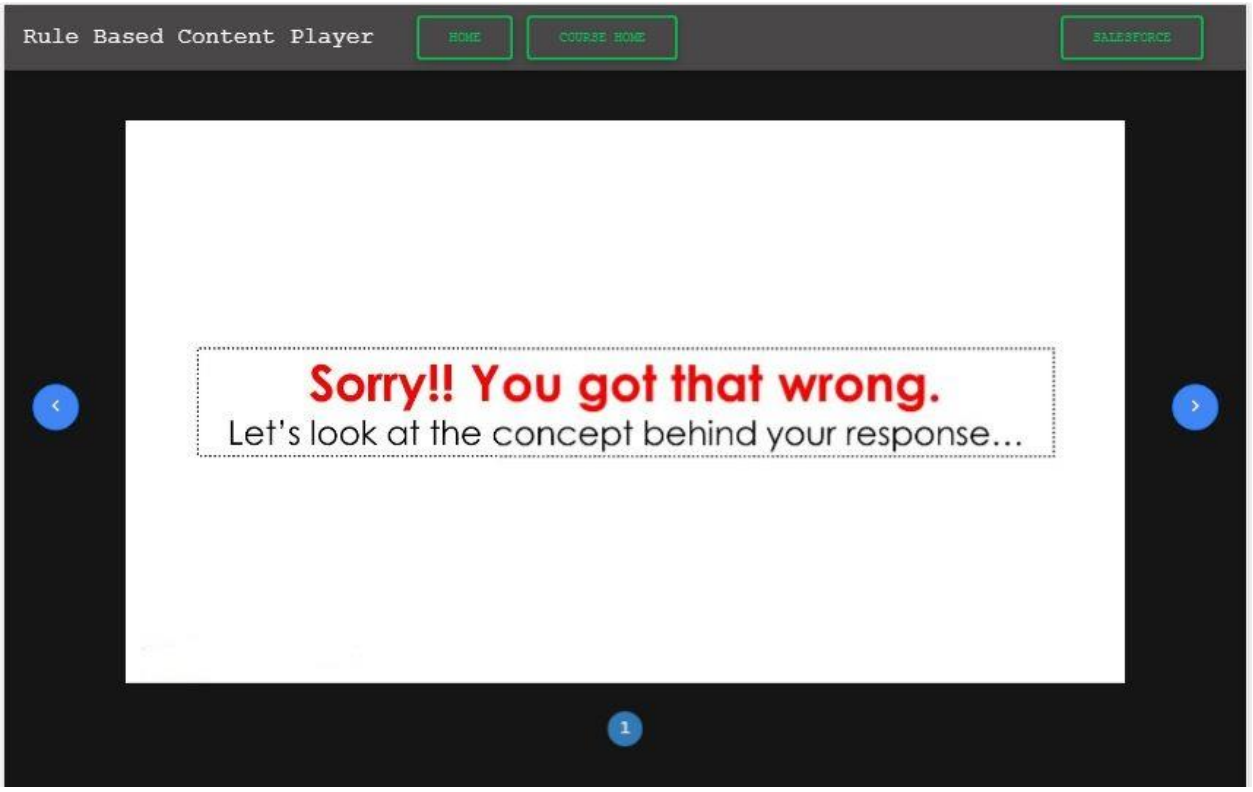
5.1.3 Quiz resource for new course



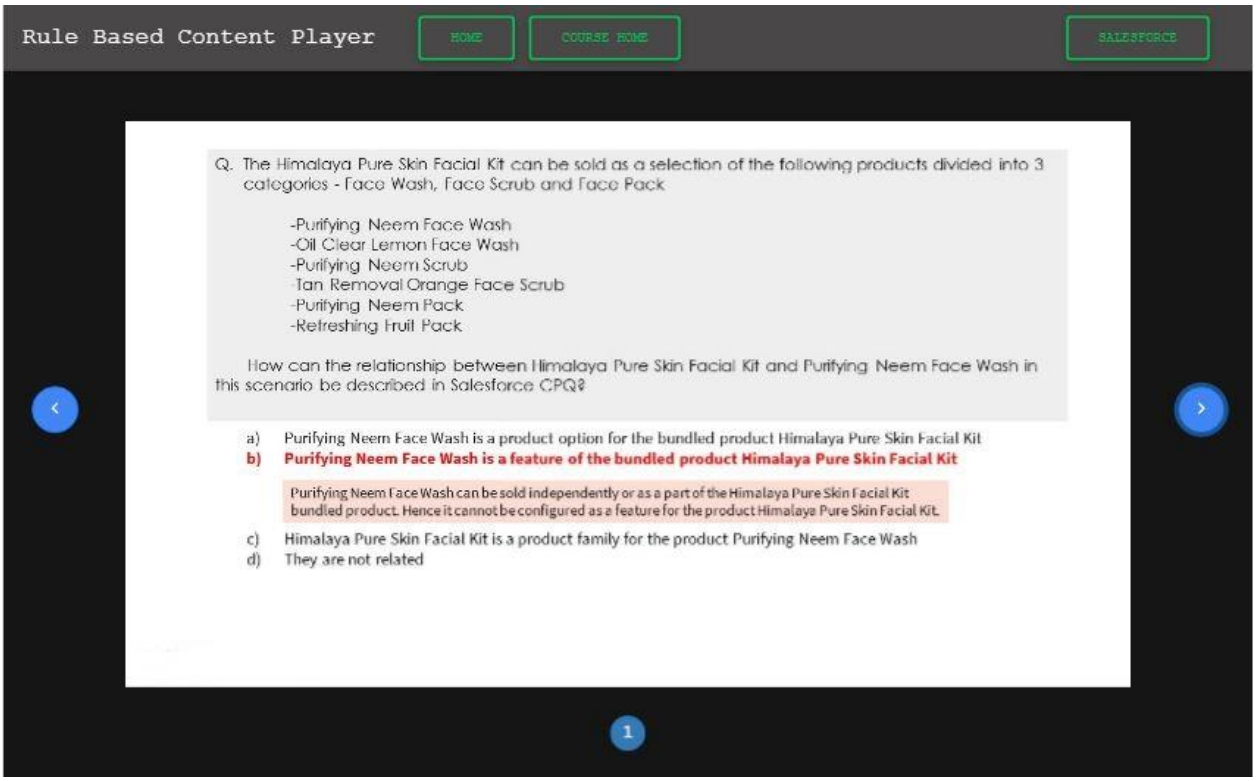
5.1.4 Resource displayed when the answer is correct.



5.1.5 Resource displayed for correct answer explanation.



5.1.6 Resource displayed when the answer is wrong.



5.1.7 Resource displayed for wrong answer explanation.

CONCLUSION

Rule Based Content Player is a personalized learning platform, which helps the user to learn the course in a better way. The player shows the content that is relevant to the user so that he can save the time that is spent in reading unnecessary content. The application is a kind of abstraction so that the application work perfectly for different levels of users. The educators can make the rules and the player shows the content according to the way that mentioned in the rule. The course flow is controlled by the rule so that well-formed rule is necessary.

FUTURE ENHANCEMENT

The future scope of this project are as follows

- Improve the content management using machine learning.
- Can implement recommendation system that helps the user to learn the related courses.

BIBLIOGRAPHY

WEBSITES

1. www.angular.io
2. www.w3schools.com
3. www.wikipedia.org
4. www.material.angular.io
5. www.ngrx.io
6. www.youtube.com
7. www.toddmotto.com

15BIT029.docx

ORIGINALITY REPORT

5%	5%	1%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	rajvishah.weebly.com Internet Source	2%
2	www.coursehero.com Internet Source	<1%
3	www.cse.iitd.ernet.in Internet Source	<1%
4	trace.tennessee.edu Internet Source	<1%
5	www.diva-portal.org Internet Source	<1%
6	www.storage4grid.eu Internet Source	<1%
7	"Proceedings of the International Conference on Information Engineering and Applications (IEA) 2012", Springer Nature, 2013 Publication	<1%
8	www.ciu.cqu.edu.au Internet Source	<1%

9	bscw.fi-ppp.eu Internet Source	<1%
10	www.scribd.com Internet Source	<1%
11	cheap.hostshield.com Internet Source	<1%
12	pt.scribd.com Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		