**Project Specification Document**

1. **Project and Developer details:**
   1. **Project details:**

* Welcome page with number of question and quiz rules:
* Text box for enter your name
* Start quiz button to start the quiz
* Once you click start quiz button it will redirect to actual quiz page:
* Shows point for your score.
* Shows all number of questions how many you must attempt.
* Shows options for timer have 60 sec for each question to answer
* Question with answers.
* Previous question, Refresh and next question button are available.
* The point will increase or decrease based on answer.
* Right answer gives you +10 points and show you green button.
* Wrong answer gives you -10 points and show you red button.
* List of questions shows you how many questions will get in quiz as well showing the current question number.
* The Timer option provide you 60 sec each question to answer if you not answer within 60 sec controls automatically move to next question and deducted 10 points.
* Question with answer section provide your current question with all possible option.
* If clicked option is right then its shows in green and give you +10 points.
* If clicked option is wrong then its shows in red and give you -10 points.
* Previous, Refresh and next button:
* Previous button go back to old question which you just attempted.
* Next button sent you to next question which will attempt.
* Refresh Button refresh your quiz and reset start from question 1 as well point also reset.
* The Result page shows once you completed your Quiz at the end it showing you Congrats message and your result.
* Total question attempted.
* Total correction and wrong answer
* Your final score.
  1. **Developer Details:**

Application is developed by Chhaya Askar and As a Full Stack Developer, I have designed and developed online Test Application system that enables users to provide online tests, review them, and display the results. The name of my application is “**Quiz Application**”

1. **Sprint Planning:**

**task:1** Create angular Project

Generate Header Component

1)app.component.CSS:

2)app.component.HTML:

* activate Route use <router-outlate>
* include <app-header>

3)app.component.ts:

4)app-routing.module.ts:

* {path:'welcome’, component:WelcomeComponent}
* {path:'question', component:QuestionComponent}

5)app.module.ts : import HttpClientModule

6)assets: keep your JSON format question file

**task:2** Generate Welcome Component

1)Add Html

* create class for welcome message and quiz rules
* provide text box to Enter Your Name to start quiz.//provide id to enter name and use same id in question component
* on start quiz button provide routerLink = "/question" and called startquiz() by using event to move control to next page.

2)Add CSS:

3)add TS:

* use annotation with id i.e. @ViewChild ('name')
* Create Start quiz () to start the quiz

**task: 3** Generate Question Component //ng g c question

1)Add html

* create class to display Welcome user message
* create class for print the point of your quiz, remain question and timer.
* create class for progress bar
* Create class for question and answers
* Create buttons to go back and , forward and refresh.
* Previous question
* Refresh
* Next question

2)Add CSS:

3) Add ts:

* to get name from welcome component use “localStorage.getItem("name")”
* getAllQuestion(){} and constructor(private questionService:QuestionService){}
* declare questionList, curruentQuestion, point and counter;
* nextQuestion(){} and privousQuestion(){}
* answer(currentQno:number,option:any(){}
* startCounter(){},stopCounter(){},resetCounter(){},resetQuiz(){},getProgressPecent(){}

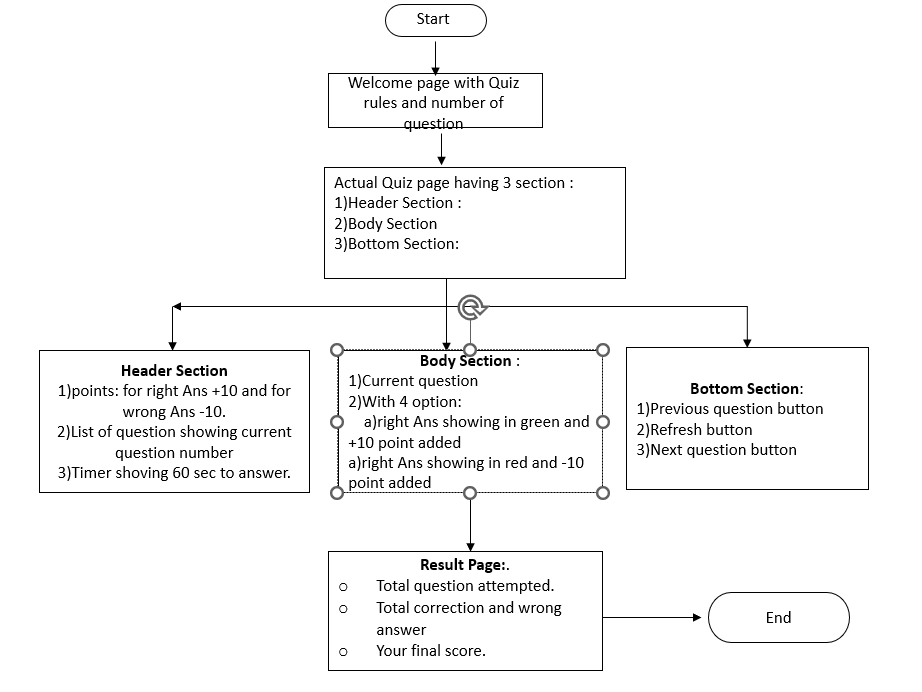
4)Use <ng-container> to display the result.

* display congratulation message and display the result.
* Total question attempted.
* Total correction and wrong answer
* Your final score

**task 4:** Creating service to access the JSON format question data into the question component

* question.service.ts :
* import the HttpCLient module and constructor (private http:HttpClient){}
* generate getQuestionJson(){}

1. **Flowchart:**

****

**4. Core concepts used in the project Concepts used in developing the application are**

1. Visual Studio Code IDE
2. HTML
3. CSS
4. Bootstrap
5. Angular JS
6. Type Scripting

**5. Links to the GitHub repository to verify the project completion**