H2: SOFTWARE TECHNICAL SPECIFICATION TEMPLATE

[CLICK HERE TO MAKE A COPY]

PROJECT NAME: Wasla

AUTHOR: Abdullah Al-Sqoor

DATE: 10-1-2020

Version 1.0.0

REVISION HISTORY			
DATE	VERSION	DESCRIPTION	AUTHOR

Table of Contents

INTRODUCTION

PURPOSE

Identify and describe scope of product whose technical specifications are being documented, and describe desired outcome.

A web application that suppose to teach the English speakers that are willing to learn the Arabic language, just exercises that teaches them some popular phrases the the alphabets, in a gamific approach, so it will be as game or exercises drills they will be doing in order to learn Arabic phrases and words.

INTENDED AUDIENCE AND PERTINENT SECTIONS

Describe each type of reader, including developers, users, testers, writers, marketing, etc., and outline which portions of document are crucial to their department.

This document should be available for the ones who willing to review this project (Coding Academy or Simplon Jury whom they are willing to valuate the project as Master Piece (Final Project) as part of the Orange Jordan Coding Academy.

PROJECT SCOPE

Describe relevant benefits, objectives, and goals and how they relate to corporate goals and strategies.

This is final project (Master Piece) as part of Orange Jordan Coding academy Final Project, This project suppose to teach the English speakers the Arabic language, and this suppose to be the first sprint.
This web application has some drills the targeted users will practice on them to strengthen their language skills and help them memorize the phrases easier as they would be engaged will learning.

DOCUMENT CONVENTIONS

Describe any naming or structural conventions employed throughout document and how they benefit reader.

MERN Stack: acronyms MongoDB database, ExpressJs backend framework, ReacJs Front-end framework and Nodejs, is a bunch of technologies to build this web application.

MongoDB: MongoDB is a cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schema.

NodeJs: is an open-source, cross-platform, JavaScript runtime environment that executes JavaScript code outside of a browser

Express.js, or simply **Express**, is a web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs.

React (also known as React.js or ReactJS) is a JavaScript library[3] for building user interfaces. It is maintained

by Facebook and a community of individual developers and companies.

JWT: acronyms JSON Web Token; is an Internet standard for creating JSON-based access tokens that assert some number of claims. For example, a server could generate a token that has the claim "logged in as admin" and provide that to a client. The client could then use that token to prove that it is logged in as admin.

JSON: JavaScript Object Notation is an open-standard file format or data interchange format that uses human-readable text to transmit data objects consisting of attribute–value pairs and array data types (or any other serializablevalue).

API: An **application programming interface** (**API**) is an interface or communication protocol between different parts of a computer program intended to simplify the implementation and maintenance of software.

Fron-end Back-end:
In software engineering, the terms **front end** and **back end** refer to the separation of concerns between the presentation layer (*front end*), and the data access layer (*back end*) of a piece of software, or the physical infrastructure or hardware.

JavaScript often abbreviated as JS, is a high-level, just-in-time compiled, multi-paradigm programming language that conforms to the ECMAScript specification. JavaScript has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

A **NoSQL** (originally referring to "non SQL" or "non relational")[1] database provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases.

REFERENCES

List any referenced document names or links.

https://en.wikipedia.org/wiki/JSON
https://en.wikipedia.org/wiki/JSON_Web_Token
https://en.wikipedia.org/wiki/MongoDB
https://en.wikipedia.org/wiki/Express.js
https://en.wikipedia.org/wiki/Node.js
https://en.wikipedia.org/wiki/React_(web_framework)
https://en.wikipedia.org/wiki/Front_and_back_ends
https://en.wikipedia.org/wiki/Application_programming_interface
https://www.json.org/json-en.html
https://jwt.io/
https://www.mongodb.com/
https://expressjs.com/
https://nodejs.org/en/
https://reactjs.org/
https://www.geeksforgeeks.org/mern-stack/
https://en.wikipedia.org/wiki/JavaScript
https://developer.mozilla.org/en-US/docs/Web/JavaScript
https://en.wikipedia.org/wiki/NoSQL
https://www.mongodb.com/nosql-explained

DESCRIPTION

PRODUCT PERSPECTIVE

Provide context and origin of product, along with expected functionality.

As mentioned earlier Wasla is a web application for teaching English speakers The Arabic Language while they playing or solving exercises, so Questions will be provided to the learners and they shall be answering them, the exercises provided in many modules (lessons).

After the leaner user login they will see list of lessons they can pick one of them and after that they will see questions and they shall answer after that they would be notified if their answer was correct or incorrect with providing the correct answer if they missed later no they'll be passed another questions until they finish them all, and if the miss answering some questions the questions will be repeated.

They questions shall be easy and have phrases in Arabic the main goal of them to teach these phrases to the learners, it can be audio text or video.

Another section of the project is for content managers which can add lessons and add the exercises for each lesson.

FEATURES

List main features with brief description.

Registration: sign-up and login as a learner (normal user) after register the user would access the lessons.
List of Lessons: drills and exercises that that user will use. Each lesson would have some exercises that specific on a subject.
Registration: sign-up and login as an admin (content manager) the person who add the lessons and drills for the learners.

USER OVERVIEW

Define groups, and describe user characteristics.

Learner: normal users who speaks English and want to learn Arabic language on our website.
Content manager: the person who add the lessons and drills on the website.

OPERATING ENVIRONMENT

Illustrate where software will function: hardware platform, operating systems, and other software components or applications with which it must work in conjunction.

Operation system for the application to run as server: Windows, Mac or Linux (it been testes on Linux mainly)
NodeJS: run time environment to run the JavaScript on the server. Which is needed to run the ExpressJS Framework for the back-end(API) and ReactJS Framework for the Front-end
MongoDB server on the server or as remote service on atlas or mlab.com (which a website provide MongoDB servers online.

For the users of this website, it need laptop/computer and a web browser (Google Chrome, Firefox, … etc).

CONTRAINTS: IMPLEMENTATION / DESIGN

Describe limitations impacting development.

Only learners can go to the lessons whenever they authenticated as logged in users
Only content managers can go to the admin dashboard and add more lessons or update them whenever they are authenticated logged in as content managers.
Learners can be registered once with an email.
Not anyone can register as a content manager.

DOCUMENTATION

Describe content, mode of delivery, and standards.

Content Manager User:
After login you you'd see the dashboard which you can click add new button to add a new lesson or choose one of the previous lessons if there were any and updated the previous question or add new one to the same lesson.

Learner User:
After Sign up/ login you should see the lessons page where you can select one lesson and after that the drills (exercises) typically questions and will choose from and it will give you the result if it was correct or incorrect, and will keep answering questions until you finish them, however if you do not answer correctly the question will be repeated at the end of the questions.

ASSUMPTIONS / DEPENDENCIES

Detail all assumed factors (not known facts) that could potentially impact technical specifications set forth. Include external factors.

NodeJS version 10.16.0
MongoDB version 4.2.2
ReactJS version 16.11.0
ExpressJS version 4.17.1
Development Operation System Ubuntu 18.0.4.3

SYSTEM FEATURES

SYSTEM FEATURE 1

DESCRIPTION AND PRIORITY	Register system for learners – high priority
STIMULUS / RESPONSE SEQUENCES	Stimulus: User clicks on Login Link. Response: Login Page is displayed Stimulus: User Enters Username and Password Response: Username and Password are validated from Database. Stimulus: User Clicks on Login Button Response: Home Page is displayed if Username and Password is correct else Error Message is displayed.
FUNCTIONAL REQUIREMENTS	1: The user shall be able to view and click on Login Link. 2: The user shall be able to enter the username and password 3: The database shall be able to validate username and password.

SYSTEM FEATURE 2

DESCRIPTION AND PRIORITY	View Lessons - Provide the user with a page to view lessons and to view exercises associated with each Lessons – high priority
STIMULUS / RESPONSE SEQUENCES	Stimulus: User clicks on Lesson Link Response: lessons are displayed Stimulus: User Clicks on a particular lesson Response: The course page and associated excises are displayed.
FUNCTIONAL REQUIREMENTS	1: The user shall be able to view and click on the Lesson Link. 2: The user shall be able to view the exercise associated with each lesson.

SYSTEM FEATURE 3

DESCRIPTION AND PRIORITY	Attempt Question Exercises - The user (learner) shall be able to attempt the Exercises.
STIMULUS / RESPONSE SEQUENCES	Stimulus: User clicks on one of the exercises answers Response: The correct answer is displayed along with incorrect and the selected answer. Stimulus: User clicks on Next Response: The subsequent quiz page is displayed. Stimulus: User clicks on Finish Quiz Response: The quiz result is displayed and go to the lessons page to choose from another lesson.
FUNCTIONAL REQUIREMENTS	The user shall be able to view and answer the exercise questions.

REQUIREMENTS OF EXTERNAL INTERFACE

USER INTERFACES
Describe product / user interface characteristics, including standards, style guides, constraints, functionality, and sample screens if applicable.

Login page and register for Learners page
Lessons List page
Exercises page
Report page after finishing a lesson
Admin Login page
Admin dashboard where can add new lesson
lesson page where can add new exercises and phrases upload videos

Database Design

It shall be a NoSQL so there's no relation specific.

HARDWARE INTERFACES
Describe product / hardware interface characteristics, nature of interactions, and communication protocols.

Processor 1GHz or faster
RAM 1GB (32bit) or 2GB (64bit)
Hard disk space 16GB (32bit) or 20GB (64bit)

SOFTWARE INTERFACES
Describe product / other software interface characteristics, including component names and versions, databases, operating systems, libraries, tools, etc. Specify any constraints, along with nature of communications and what data is coming in and being disseminated.

MongoDB 4.4.0 and NodeJS 10.16.0 must be installed

COMMUNICATION INTERFACES
List requirements of communication functions: email, browsers, servers, forms, etc. Describe standards employed, security or encryption measures, data transfer rates, and synching.

The system shall use MongoDB Database named "wasla".
The system uses NodeJS 10.16.0 For API & front-end framework (ReactJS).

ADDITIONAL NONFUNCTIONAL REQUIREMENTS

PERFORMANCE

Specify performance requirements. For additional clarity, provide rationale.

Responses to queries shall take no longer than 3 milliseconds to load onto the screen after the user submits the query for 1 user.
The system shall display confirmation messages to users within 4 milliseconds after the user submits information to the system.
The system should generate policy with an accuracy of 99%.

SAFETY

Specify safety requirements. Define safeguards and systems employed. Identify external policies and regulations impacting safety requirements.

Consistency: Checking the fact that all clients must be attached to one server, so there is an appropriate control of the information.

SECURITY

Specify security requirements. Define safeguards and systems employed. Identify external policies and regulations impacting safety requirements.

The admin dashboard (content manager) should be available for these type of users and not any one can register to be a content managers.

Only content manager shall add lessons and update them.

The learners cannot see the lessons unless they are authenticated and logged in.

SOFTWARE QUALITY

List other characteristics crucial to success of product. List each by describing its relation to product, being quantitative, specific, and verifiable.

Availability-1 The system shall be available to users all the time.
Availability -2 The system shall always have something to function and always pop up error messages in case of component failure.
Efficiency-1: The system shall generate the correct pages with an accuracy of 99%. Efficiency-2: The system shall provide the right tools to support all its features.

APPENDICES

APPENDIX A: GLOSSARY OF TERMS

Define all terms and unique acronyms employed throughout document and specific to project.

APPENDIX B: ANALYSIS DOCUMENTATION

List file / document names / provided links to all diagrams, models, additional findings pertinent to technical specification development.

APPENDIX C: ISSUES

List all unresolved issues, TBDs, pending decisions, findings required, conflicts, etc.

ISSUES		
ID	**DESCRIPTION**	**PARTY RESPONSIBLE**