



CAPSTONE PROJECT 2

CMU-SE-451 / CMU-IS-451 / CMU-CS-451

TECHNOLOGIES STACK DOCUMENT

Version 2.0

Date: 1 - Mar - 2021

EXPERT-DRIVEN SMART DASHBOARD APPLICATION

Submitted by

Vo Van Hoa
Pham Van Tin
Ky Huu Dong
Tran Thi Thanh Kieu

Approved by

Capstone Project 2 - Mentor:

Name

Signature

Date

Binh, Thanh Nguyen _____

A handwritten signature in blue ink, appearing to be 'Nguyen Binh', written over a horizontal line.

_____ 26 - May - 2020

PROJECT INFORMATION			
Project Acronym	EDSDA		
Project Title	Expert-Driven Smart Dashboard Application		
Project Web URL	https://sda-research.ml/		
Start Date	01 - Mar - 2021		
End Date:	02 - Jun - 2021		
Lead Institution	International School, Duy Tan University		
Project Mentor	Ph.D Binh, Thanh Nguyen		
Scrum Master	Hoa, Vo	hoavo.dng@gmail.com	0935.193.182
Team Members	Tin, Pham Van	tinphamvan123@gmail.com	0932.535.175
	Dong, Ky Huu	kyhuudong@gmail.com	0898.246.980
	Kieu, Tran Thi Thanh	thanhkieuTRAN391@gmail.com	0358.583.251

DOCUMENT INFORMATION			
Document Title	Technologies Stack		
Author(s)	Team C2SE.06		
Role	[EDSDA] TechonologiesStack_v2.0		
Date	01 - Mar - 2021	Filename	[EDSDA] 012 Technologies Stack
URL	https://github.com/sdateamdtu2020/SDA-v2.0		
Access	Project and CMU Program		

REVISION HISTORY

Version	Person(s)	Date	Description	Approval
Draft	Hoa, Vo	01 - Mar - 2021	Initiate document	x
2.0	All members	26 - May - 2020	Finish content of document	x

TABLE OF CONTENTS

PROJECT INFORMATION	1
DOCUMENT INFORMATION	1
REVISION HISTORY	2
TABLE OF CONTENTS	3
1. INTRODUCTION	5
1.1. PURPOSE	5
1.2. DOCUMENT OBJECTIVES	5
1.3. INTENDED AUDIENCE	5
1.4. ACRONYMS AND ABBREVIATIONS	5
2. MAIN DECISIONS TO CHOOSE TECHNOLOGIES	5
3. PROGRAMING LANGUAGE	6
3.1. JAVASCRIPT	6
3.2. PYTHON	6
3.3. HTML/CSS	6
3.4. SQL	6
4. TECHNOLOGIES OF WEB APPLICATIONS	6
4.1. REACT	6
4.2. REDUX	7
4.3. RESTful API	7
4.4. ExpressJS	7
5. Database / Data Cubes....	8
5.1. PostgreSQL	8
5.2. PgAdmin4	8
5.3. RML.io	8
5.4. GraphDB	8
6. DEVELOPMENT STACK	8
6.1. VISUAL STUDIO CODE	8
6.2. ESLINT	9
6.3. PRETTIER	9
6.4. REACT DEVTOOLS AND REDUX DEVTOOLS	9
7. VERSION CONTROL	9

7.1. Bitbucket	9
8. DEPLOYMENT	9
8.1. FIREBASE	9
8.2. VIRTUAL MACHINE	10
8.3. Apache Airflow	10

1. INTRODUCTION

1.1. PURPOSE OF DOCUMENT

The purpose of the Technology Stack Document is to provide a brief, yet clear information about all the technologies that make the project.

1.2. DOCUMENT OBJECTIVES

The Technology Stack has the following objectives:

- To describe the technologies that we are using to build the project.

1.3. INTENDED AUDIENCE

This document is intended for the following audiences.

1.4. ACRONYMS AND ABBREVIATIONS

Acronym/Abbreviation	Meaning
EDSDA	Expert-Driven Smart Dashboard Application

2. MAIN DECISIONS TO CHOOSE TECHNOLOGIES

We have some key decisions to choose our technologies stack:

- Open source technologies: We would like to choose all the frameworks, tech stacks that are licensed under MIT license so that we can have more support from the community
- New but proven technologies: We use React, Redux as well as NodeJS for our project because they are new, so that we can really learn new things by building our project. Also, they've been used in production in many projects by the community so that we know that they are safe.
- Document rich: We want to have a framework with a good and well organised document so that we save our time reading the docs.

3. PROGRAMING LANGUAGE

3.1. JAVASCRIPT

JavaScript often abbreviated as JS, is a high-level, dynamic, weakly typed, prototype-based, multi-paradigm, and interpreted programming language. It's the most popular language nowadays and can be used everywhere. Javascript is so popular that we can easily find a solution for our problem on Google and Stackoverflow.

3.2. PYTHON

Python is an elegant, versatile language with an ecosystem of powerful modules and code libraries. There are over a hundred tools that act as a framework, libraries, or software for ETL. These tools become your go-to source once you start dealing with complex schemas and massive amounts of data. Therefore, we use Python to make the ETL process become easier to program and more effective.

3.3. HTML/CSS

HTML (the Hypertext Markup Language) and CSS (Cascading Style Sheets) are two of the core technologies for building Web pages. HTML provides the structure of the page, CSS the (visual and aural) layout, for a variety of devices. Along with graphics and scripting, HTML and CSS are the basis of building Web pages and Web Applications.

3.4. SQL

HTML (the Hypertext Markup Language) and CSS (Cascading Style Sheets) are two of the core technologies for building Web pages. HTML provides the structure of the page, CSS the (visual and aural) layout, for a variety of devices. Along with graphics and scripting, HTML and CSS are the basis of building Web pages and Web Applications.

4. TECHNOLOGIES OF WEB APPLICATIONS

4.1. REACT

Home page: <https://reactjs.org/>

React is the current industry standard that offers a lot of out of the box benefits. It is fast, efficient, and scalable. Due to the large community, finding solutions to potential problems

and reference material is much easier, even for a potential dev without a lot of experience who would like to contribute to the Main Course.

Pros of React

The community

Like with most online developer communities, the React one is growing and offers a great network of experienced developers.

Faster development

Speed is often the name of the game where development is concerned. Anything that can speed up the development of an application is much appreciated!

With React, the development time is considerably shorter.

4.2. REDUX

Home page: <https://redux.js.org/>

Redux is a predictable state container for JavaScript apps.

It helps you write applications that behave consistently, run in different environments (client, server, and native), and are easy to test. On top of that, it provides a great developer experience, such as live code editing combined with a time traveling debugger.

You can use Redux together with React, or with any other view library.

4.3. RESTful API

RESTful API is an architectural style for an application program interface (API) that uses HTTP requests to access and use data. We build APIs based on RESTful principles for easy data accessing and interacting between different levels of the system.

4.4. ExpressJS

A prebuilt NodeJS framework that makes creating server side applications simple, fast, and flexible. NodeJS is powered by Google's V8 Engine which means it's powerful and can handle a large number of requests without lapsing in dependability. Also, this means that this is a highly scalable choice when you consider the Event Loop which manages all asynchronous operations allowing the program to continue to run as expected without stops.

5. Database / Data Cubes

5.1. PostgreSQL

PostgreSQL is one of the most well developed and mature relational database management systems, with its development, ongoing for more than 20 years now, from supporting small websites on single machines to data warehouses and large scale concurrent setups.

5.2. PgAdmin 4

pgAdmin is the most popular and feature rich Open Source administration and development platform for PostgreSQL, the most advanced Open Source database in the world.

5.3. RML.io

To execute the process of generating RDF Data Cubes from Mapping Rules and data sources, we use the RMLMapper and RMLStreamer frameworks provided by the RML.io organization. They are platforms that help users produce high quality linked data and knowledge graphs that are fast and easy to install.

5.4. GraphDB

Enterprise RDF and graph database with efficient reasoning, cluster and external index synchronization support.

6. DEVELOPMENT STACK

6.1. VISUAL STUDIO CODE

Home page: <http://code.visualstudio.com/>

Visual Studio Code (Vscode) is a great open source code editor developed by Microsoft, it provides a rich and easy to use user interface as well as thousands of extensions that we can install to suit our needs.

6.2. ESLINT

Eslint is a must have tool to lint our project, so that we can provide a consistent code style to work in our team.

6.3. PRETTIER

Prettier, used with Eslint, helps us to auto format our code on save.

6.4. REACT DEVTOOLS AND REDUX DEVTOOLS

React and Redux are being used heavily in our project. That's why the devtools will help us debug our code easier.

7. VERSION CONTROL

7.1. Bit Bucket

Home page: <https://bitbucket.org>

Bitbucket is a repository hosting service. Think of it as the "cloud" for code.

The main benefit of Bitbucket is storing the code in one and secure place, leaving it open for all teammates, so they can without any obstruction get access to the code or parts of it, or pull out the code based on their work. Also you can easily track all changes and rollback if needed. Actually, you can store not only code but also the scripts and even the server configuration. Due to the simple and straight-forward web interface, you can navigate and search for what you need without any effort.

8. DEPLOYMENT

8.1. FIREBASE

Home page: <http://firebase.google.com/>

We use Firebase to deploy hosting

Firebase, offered by Google, is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search and YouTube. Alongside a set of management tools, it provides a series of modular cloud services including computing, data storage, data analytics and machine learning.

Firebase provides a rich set of tools, configs and tutorials that can help us quickly deploy our application in just a matter of minutes.

8.2. Virtual Machine

Linux VPS is a cost methodology in which there is a physical server, although several completely independent virtual servers run from each other. We use Linux VPS to configure server configurations in the cloud, as well as a cloud environment for deploying and launching APIs, thereby making it easier for users to interact with the system and database.

8.3. Apache Airflow

Apache Airflow is an open-source workflow management platform. The solution is to manage the project's increasingly complex workflows. Creating Airflow allowed programmatically author and schedule the workflows and monitor works via the built-in Airflow user interface.

Airflow is written in Python, and workflows are created via Python scripts. Airflow is designed under the principle of "configuration as code". While other "configuration as code" workflow platforms exist using markup languages like XML, using Python allows developers to import libraries and classes to help them create their workflows.