January’s Reflective Journal

**Student Name:** Daniel Costel Neagu

**Student ID:** X17128463

**Programme:** BSc (Honours) in Computing - Software Development

**Project Title:** *Farmers Market.ie*

# Week 15: 4th January / 10th January

This week focus is to review the project stack to be able to understand it better and make sure I employ the right technologies to develop my product on. When I started this project I only knew that is going to be an E-Commerce website application, and with the technologies taught in college during the past three years it will come in handy getting my information from to help me build a successful application.

My focus is on MERN architecture which is a Full-Stack architecture, as it allows me to easily construct a tree tier project (Front-end, Back-end and Database) using JavaScript and JSON all the way through.

For the Front-end tier, my MERN stack project is used React.JS, for the middle tier (Back-end/server-side) I will use Express.JS and Node.JS and for the last tier of the framework, I will use MongoDB as my Database.

This week I managed to collect all the material necessary regarding the technologies that I will use in my project from now on and I managed to review them successfully to undescended better how to work with them.

Next week focus is to start creating the database model that will hold data about users and admin with their afferent rights and attributes. Another task for next week is to meet up with the project supervisor to update him on my project progression.

# Week 16: 11th January / 17th January

This week focus is to work on the Database side of the project and get in touch with my faculty designated project supervisor over the team’s video call chat meeting.

Today is Thursday and I have met up with my project supervisor to talk about my progression made so far on the project. The meeting went well, and I have received some information regarding my mind term submission on the project. During the week I have started to work on creating an account on MongoDB and deploying it on Amazon Web Services Cloud Provider side and to be able to perform get and post actions on the database I started using Postman as an API. I am using Mongoose, which is a MongoDB object modelling tool designed to work in an asynchronous environment and it supports both promises and call-backs. Mongoose provides a straight-forward, schema-based solution to my application data model. It includes built-in type casting, validation, query building, business logic hooks and more, out of the box.

For this part of the project, I had to initiate the project to create the JS packages and create the server with the database schema. Other technologies used and installed are express validator and nodemon, I had to import dontenv library to be able to use .env

Next week focus is to start working on creating and adding on the model Database side for the Products Categories, Produces and to create a Shopping Cart to hold Produces.

# Week 17: 18th January / 24th January

This week focus is to add more data in my Database by creating Produces model, Category model, and the Cart.

Product creation was made on the back-end with the help of multer npm to use on the DB side: npm install –save multer and with npm install –save shortid to save files.

Creating the Category sections in the back-end on DB and I am using Slugify library: npm install -- save slugify. The Shopping Cart was created too and it can hold produces added by the user after they are signed in, the admin now can add produces and categories in the DB.

Next week focus is to start my React.JS application for the Admin dashboard and meet up with my project supervisor over the teams.

# Week 18: 25th January / 31st January

This week focus is to work on my React.JS application for the Admin dashboard, the front end and connected it with the back end that was created last week.

I created a new React app for the project called Admin Dashboard and with the help of Route as a logic component, I will be able to navigate to the Home, Register and Login pages. Then I installed redux, thunk libraries and to be able to handle API calls as a centralized API I used axion and to allow all URLs in between my apps(like back end and front end for the moment) I used cors library.

The Admin Dashboard connects to the back end and authenticates the user and admins.

The meeting hold on Wednesday with my project supervisor went well and he suggested that I separate access to the admin dashboard by using user categories.

Next week focus it will be in adding more functionality to the admin dashboard.