ICE Malta

MySuccess Website Developer

Final Project v2.3

icemalta.com

ICE Malta

MySuccess Website Developer

Final Project v2.3

Contents

Project Overview

This project is designed to exercise your skills on multiple concepts covered throughout the MySuccess Website Developer course. You will be building a product registration system for a smart-appliance company called 'Kahuna'.

Skills you will exercise in this project include:

- Designing a data storage model (using an ERD).
- Creating and implementing a database.
- Creating a REST API backend using PHP.
- Testing an API.
- Making a project available via source control and creating a README file.

The exact methodology of designing your system is up to you! Check the following sections to see what you need to build.

Final Project

Kahuna Control

Scenario

Kahuna Inc. is a manufacturer of smart home appliances. They wish to implement a portal where customers can register their appliances. Once registered, customers will be able to check on their products, and see information such as serial number, purchase date, warranty time left and so on.

For this prototype system, the following products should be made available to register:

Table 1: Product Database

Serial #	Product	Warranty
KHWM8199911	CombiSpin Washing Machine	2 Years
KHWM8199912	CombiSpin + Dry Washing Machine	2 Years
KHMW789991	CombiGrill Microwave	1 Year
KHWP890001	K5 Water Pump	5 Years
KHWP890002	K5 Heated Water Pump	5 Years
KHSS988881	Smart Switch Lite	2 Years
KHSS988882	Smart Switch Pro	2 Years
KHSS988883	Smart Switch Pro V2	2 Years
KHHM89762	Smart Heated Mug	1 Year
KHSB0001	Smart Bulb 001	1 Year

Deliverables

For this project, you will need to deliver:

Deliverable 1: REST API

A REST API backend, written in PHP. The API should have the following endpoints:

Non-Authenticated Endpoints

The following endpoints do not require a user to be logged in.

- **Create Account** – this should allow an account to be created, either as a 'client', or as an 'admin'.
- **Login** – allows users to login.

Authenticated Endpoints

The following endpoints should require a user to be logged in.

- **Register product** – this should allow clients to register products which they have purchased. The serial number for the product should be one of the numbers provided in Table 1: Product Database.
- **View products** – this should allow clients to view a list of products they have registered.
- **View product** – given a product serial number, this allows a client to view details such as the product name, and how much warranty they have left on this product.
- **Logout** – this should allow a client to log out.

Admin-Only Endpoints

These endpoints should only be available to users logged in as 'admin'. Note that admins can also use any of the endpoints above.

- **Add product** – allows an admin to add a new product, which can then be registered by clients.

Deliverable 2: Database

A database, including designs (ERD), with populated data. This design should accommodate all the data storage requirements outlined in this assignment.

Deliverable 3: API Tests

A collection and environment for testing the API with Postman or a similar tool. Each of the endpoints from Deliverable 1 should have an associated test in Postman.

Deliverable 4: Distribution

- Instructions on setting up and running your project in a README.md file.
- A link to a GitHub repository.

Architecture

Deliverable 1: REST API

The architecture of this project is up to you. You can use a Docker environment as we did in class, or you can use any other method such as a local installation, cloud-service, PHP integrated web server etc. For the authenticated endpoints, you are free to use the same token code we used in class or implement any other simple authentication mechanism.

Deliverable 2: Database

You are free to use any DBMS for this – whether relational (MariaDB, MySQL, Oracle, SQL Server…) or non-relational (MongoDB, Cassandra, Couchbase) etc. The requirement is for all data needed for the project to be stored in the database, and for an Entity Relationship Diagram (ERD) to be provided.

Deliverable 3: API Tests

You should deliver a testing suite which allows for easy testing of your API. This can be done via Postman, or any other similar technology such as HTTPie, Thunder Client, Bruno or even cURL (in which case please document your endpoints).

Deliverable 4: Distribution

Please provide a README.md file with instructions on setting up your project.

You are free to upload your project to any code hosting service, as long as it is public. This includes GitHub, GitLab, BitBucket or even your own server. Your repository, however, **must** be public.

Working Environment

A working environment has been set-up for you to get started with your project. You can clone it from the following URL:

https://github.com/icemaltacode/sc-bed-finalproject-env

Using this working environment is entirely optional, but recommended if this is your first solo project.

Tip: After cloning the above repository, rename the root folder and replace ‘env’ with your name. For example: *sc-bed-finalproject-alice*.

Marking Scheme

Marks for this project are allocated as follows. **The pass mark is 50/100.**

Deliverable	Item	Marks
1	Kahuna Control backend REST API, written in PHP: • Code efficiency and structure [10]. • Separation of model and controller [5]. • Routing of requests [5]. • JSON-formatted responses [5]. • HTTP success/error codes [5]. • Well-structured model with namespaces [5]. • Fully-working code API [5].	40
2	Test suite: • Well-organised collection with complete API coverage [20]. • Use of variables to automate testing [10].	30
3	Database design and implemented database. • Well-structured ERD [10]. • Database implementation matching ERD [10].	20
4	Instructions for setting up and running your project. • Code available on a public repository such as GitHub [5]. • README file using Markdown [5].	10

Important Stuff

Please take good note of the following:

- You are free to use any technologies you want in your project, however, your back end must be written in PHP. Any additional technology (such as Composer, JavaScript, Twig, front-end JavaScript frameworks and other libraries) can be used and is up to you.

- **You do NOT need to provide a front-end for your project, although you are free to do so. This, however, will not earn you any additional marks.**

- Your API does not need to be 100% working for you to pass your assignment. A full-working API is only worth 5 marks! Focus on creating good code which is easy to understand and efficient – that's where the marks are.

- Your project must be made publicly available on a code-hosting service such as GitHub, GitLab, BitBucket or similar, or on your own server.

Deadline

The exact deadline for your class will be given to you by your Educator. As a guide, this is normally 3 months from the date your assignment is presented in class.

Institute of Computer Education, Ltd.
Central Business Centre, Level 2, Suite 2,
Mdina Road,
Haz-Zebbug, Malta

+356 2146 7787
info@icemalta.com
icemalta.com