

**Internet Programming II Formative Assessment 1**

***Unit Standards covered:***

# 115431 Analyse feedback contexts and apply constructive feedback techniques

* **114050 Explain the principles of business and the role of information technology**
* ***114048 Create database access for a computer application using structured query language***
* ***115388 Produce documentation for a computer program to agreed standards***

**Instructions to Learner:**

1. Learner needs to include all criteria evidence of all tasks.
2. The work handed in must be your own and valid.
3. Please take note that failing to adhere to the criteria of each task within this assessment will lead to an immediate NYC.
4. The learner must achieve a mark of 80% to be deemed as competent

Learner Name and Surname: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Assessor Name and Surname: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Learner ID Number: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Scenario

Complete all of the following instructions. Also have a look at the mark sheet section to see how much each question counts to ensure you provide enough evidence to obtain a maximum mark.

You have to hand in a folder named Name\_Surname\_IPG512F1, which will contain all of the evidence as asked further on in this assessment.

**Phases in the project:**

Phase 1: Planning of the Project

Phase 2: Progress and first draft of application

Phase 3: Final Product and documents hand in

A company called CTU Networks is installing a complete eCommerce system for a pizza shop called The Pizzatorium. The Pizzatorium is a major supplier of pizza based in Pretoria North and they deliver as far south as Capetown and as far north as Polokwane. You have been hired by CTU Networks to create an ASP.NET MVC4 Web Application for The Pizzatorium. The website must allow users to choose the size of the pizza they want, the ingredients they want, then enter a delivery address and payment details.

Customers order pizza by first choosing a size (small R15.00, medium R25.00 or large R40.00) then adding ingredients. Finally they submit their order and give a delivery address. A total cost for the pizza must be determined. It is important that you understand the functionality of the website before you begin. It is also necessary for you to make a number of design decisions yourself, especially when it comes to choosing which control to use and where to use it. It is suggested that you draw the layout of the Views and the interaction flow before you begin coding.

Users start at a Login page, where they have a choice of logging in or creating a new user. After logging in, they proceed to the Design view, where they design their pizza. After they are happy with the pizza, they go to the ‘Final page’, where they choose payment and delivery. They also receive a breakdown of their pizza.

Finally, they are sent to the ‘Thankyou page’ where they are given an invoice, and the name of the person who will deliver their pizza.

The method for implementing data storage is entirely up to you, you either employ a Database-first approach or Code-First.

At the heart of the Database Layer, you must implement the following using EntityFramework. Note that these must be exactly as specified. Create Primary keys as necessary.

## tblUsers

This model class is used to store user information. Users have to register before they can order pizza.

Fields:

* dName (varchar 30),
* dUSerName (varchar 30),
* dAddress (varchar 50),
* dPhone (varchar 10),
* dFavPizza (varchar 30).

## tblIngredients

This model class is used to store pizza ingredients like cheese (R2.00), capers (R3.00), banana

(R2.00), avocado (R4.00), chicken (R5.00), anchovies (R5.00), sausage (R5.00) and mince (R6.00). You will have to enter data in this table for the application to work properly. Make your own decision as to how much each additional ingredient you enter costs. Fields:

* dIngredient (varchar 25),
* dCost (decimal).

## tblAreas

This class lists areas that The Pizzatorium delivers to. Enter a few areas, for example, Hatfield, Centurion, Pretoria CBD, Soshanguve. Fields:

* dArea (varchar 25).

## tblDelivery

This model class lists the delivery personnel and which area they deliver to. Enter a few people.

Ensure the areas they deliver to are the same as the areas in tblAreas.

Fields:

* dName (varchar 30),
* dArea (varchar 25),
* dPhoto (varchar (max)).

**Create the following Views and their corresponding Controllers:**

**Login** - here users can either create a new user or login. Create additional pages/view as necessary to cater for new user creation.

New records must be created for each new customer in tblUsers; users can only proceed if they enter their names and passwords.

How you handle login is up to you,it is not necessary to use the login controls; you can create your own login system easily using state control.

This is the only Web Form users can see if they have not logged in. Once they successfully login they must be redirected to the design page.

**Design** - here users choose the size of their pizzas and which ingredients to add (ingredients MUST be extracted from the database).

The size of the pizza can be chosen using CheckBoxes or a DropDown.

Ingredients can be added to the order by using CheckBoxes or a DropDown. You could create a checkbox for every ingredient, or list all the ingredients in a DropDown and have the user click a button for every ingredient they add.

Where and how you store the pizza details is up to you - you can use cookies, Session Variables, query strings or even a table in your database.

Once the customer has designed their pizza and clicked a button called Submit, they must be redirected to the ‘final page’.

**Final** - here users must be shown a breakdown of their pizza ingredients and size and the cost of the pizza.

If they decide they want to order the pizza, they can also enter a payment method and delivery area, and submit the order, using a button called Order.

If they do so, they must be redirected to the ‘thankyou page’ Use textboxes to capture additional information here.

Delivery area must be extracted from the tblAreas class in the Model.

The pizza breakdown must include size and all ingredients chosen.

If the customer wants to change their order, redirect them to the design view If the customer wants to cancel their order, redirect them to the login.

**Thankyou** - here the customer gets a thank you message and receipt from the Pizzatorium.

The cost of the pizza must be shown.

Who will deliver the pizza must be shown, name and picture (This is determined by taking the area of delivery given in the ‘Final view’ page and matching it to someone that delivers to that area in tblDelivery class). The customer's address must be shown.

**Mark Sheet**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | Phase 1 | | | | | | 0-5/10 | 6-7/10 | 8-10/10 | | | | Very little Planning noted  . The student can work towards greater planning and possible solutions before submitting the  Final Program. | Some Planning noted. Student should increase efforts in program development | Good to excellent planning shown in the development of program. | | | | Total for Phase 1: | | 10/\_\_\_ | | | | Phase 2 | | | |  | | 0-5/10 | 6-7/10 | 8-10/10 | | | | Preliminary work presented does not represent  the majority of steps in the process / Work  presented is incomplete | Preliminary work presented is adequate to track the development of the program. Certain steps in the process however are vague or absent. | Preliminary work presented clearly demonstrates the track and progress of program development | | | | Total for Phase 2: | | 10/\_\_\_ | | | | Phase 3 | | | Mark | Received | | 1) Database and all tables created, either using the code-first, model-first or database-first approach. | | | 60 |  | | 2) Your app should have a consistent layout and visual appeal using a layout view (Master page) | | | 25 |  | | 3) All views created and designed, including the login view | | | 10 |  | | 4) Create an additional view that allows to create a new user | | | 10 |  | | 5) Successfully implement navigation in the layout for the screens you designed in. | | | 5 |  | | 6) Design view - Successfully allows users to choose and create the type of pizza of choice and the ability to add ingredients, with all costs calculated and saved. | | | 20 |  | | 7) Enforcement of business rules and validations where necessary. | | | 20 |  | | 8) Final view - Is able to display the choice of pizza, with size and costs. Users have a choice to order and choose payment type and also to choose where the pizza is delivered from. Customers given an option to cancel order and returned to the login view, or change order and returned to the design view. | | | 20 |  | | 9) Thank-you view – A thank you message and receipt. View displays the picture of the delivery staff and name. It also shows the customer's address. | | | 20 |  | | 10) Technical Report document and User Manual. | | | 30 |  | | 11) Planning documents, site maps and UML diagram. | | | 20 |  | | Total for Phase 3: | | | 240 |  | | | |  |
| **Total for assessment with Bonus** | **260\** |
| **Total percentage for assessment** | **100\** |

**Competency:**

**Competent**

**Not Yet Competent**

**Final Comment**