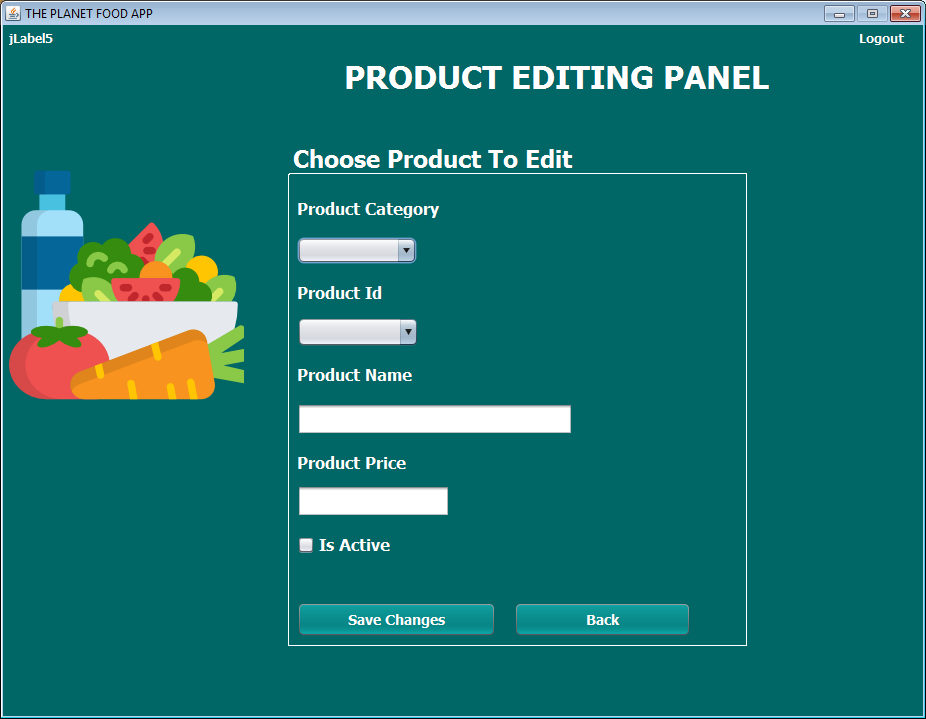
**Designing The EditProductFrame**

****

**STEPS TO BE DONE IN** **EditProductFrame**

In the **EditProductFrame** we need to do following steps:

1. Display **username** on the top left

2. Allow the user to **logout**

3. As soon as the frame loads it should fill the **Category Names** in the **JComboBox** by pulling them from the DB

4. Also the **Product Id** of the first category selected in first **JComboBox** should be automatically displayed in second **JComboBox**

5. Write code for **item selection event** in both **JComboBoxes**

6. Write code for "**Save Changes**" and "**Back**" Button

**THE TABLES USED IN EditProductFrame**

**1. PRODUCTS**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| **PROD\_ID** | **Varchar2(10)** | **Contains Id Of The Product** |
| **CAT\_ID** | **Varchar2(10)** | **Contains The Cat Id To Which This Product Belongs . It Is Also The Foreign Key** |
| **PROD\_NAME** | **Varchar2(20)** | **Contains The Name Of The Product** |
| **PROD\_PRICE** | **Number(5)** | **Contains Price Of The Product** |
| **ACTIVE** | **Char(1)** | **Contains "Y" or "N"** |

**THE POJO CLASSES USED IN EditProductFrame**

1. The **Product** POJO

**CODE FOR "Product" POJO**

***Same As Before***

**THE DAO CLASSES USED IN EditProductFrame**

1. The **ProductDao**

2. The **CategoryDao**

**HOW TO PULL CATEGORY NAMES FROM THE DB**

***To do this we will call the method getAllCategoryId( ) of the CategoryDao class***

**HOW TO PULL PRODUCT DETAILS OF ALL THE PRODUCTS OF A PARTICULAR CATEGORY**

***To do this we will call the method getProductsByCategory( ) of the ProductDao class***

**HANDLING THE CONSTRUCTOR:**

From the body of **constructor** we would the method called **loadProductDetails( )** which will load the category names in the **JComboBox** as soon as the frame opens.

***public class EditProductFrame extends javax.swing.JFrame {***

***private HashMap<String,String> categories;***

***private HashMap<String,Product> productList;***

***private String prodName;***

***private String catId;***

***private double prodPrice;***

***private String prodId;***

***private String prodIsActive;***

***private String catName;***

***private Product p;***

***public EditProductFrame() {***

***initComponents();***

***this.setLocationRelativeTo(null);***

***lblUsername.setText("Hello "+UserProfile.getUSername());***

***loadProductDetails();***

***}***

***public void loadProductDetails() {***

***try***

***{***

***categories=CategoryDao.getAllCategoryId();***

***for(String catName: categories.keySet())***

***jcCatId.addItem(catName);***

***}***

***catch(SQLException ex){***

***JOptionPane.showMessageDialog(null, "Exception in DB","Exception!",JOptionPane.ERROR\_MESSAGE);***

***ex.printStackTrace();***

***}***

***}***

**HANDLING actionPerfromed FOR jcCatId COMBOBOX**

As soon as user selects a **category name** from the **ComboBox** we need to load all the **Product Id** which belong to the **selected category** . For this we handle the **actionPerformed( )** method for the **jcCatId** **ComboBox** . From this method we call the method **getProductsByCategory( )**of the **ProductDao** class

***private void jcCatIdActionPerformed(java.awt.event.ActionEvent evt) {***

***try***

***{***

***catName=jcCatId.getSelectedItem().toString();***

***catId=categories.get(catName);***

***productList=ProductDao.getProductsByCategory(catId);***

***jcPrdId.removeAllItems();***

***for(String prodId: productList.keySet())***

***jcPrdId.addItem(prodId);***

***}***

***catch(SQLException ex)***

***{***

***JOptionPane.showMessageDialog(null, "Error while connecting to DB!","Exception!",JOptionPane.ERROR\_MESSAGE);***

***ex.printStackTrace();***

***}***

***}***

**HANDLING actionPerfromed FOR jcProdId COMBOBOX**

As soon as user selects a **Product Id** from the **ComboBox** we need to pull the selected product's details from the **productList** HashMap and load the **Product Name**, **Product Price** and **IsActive** in th UI . For this we handle the **actionPerformed( )** method for the **jcProdId** **ComboBox** .

***private void jcPrdIdActionPerformed(java.awt.event.ActionEvent evt) {***

***if(jcPrdId.getSelectedItem()==null)***

***return;***

***prodId=jcPrdId.getSelectedItem().toString();***

***p=productList.get(prodId);***

***txtPrdName.setText(p.getProdName());***

***txtPrdPrice.setText(String.valueOf(p.getProdPrice()));***

***prodIsActive=p.getIsActive();***

***if(prodIsActive.equalsIgnoreCase("Y"))***

***chkIsActive.setSelected(true);***

***else***

***chkIsActive.setSelected(false);***

***}***

**WRITING THE CODE FOR validateInput() IN EditProductFrame**

***public int validateInput()***

***{***

***int result=1;***

***prodName=txtPrdName.getText();***

***if(chkIsActive.isSelected())***

***prodIsActive="Y";***

***else***

***prodIsActive="F";***

***try***

***{***

***prodPrice=Double.parseDouble(txtPrdPrice.getText());***

***}***

***catch(Exception ex)***

***{***

***result=0;***

***}***

***if(prodName.isEmpty())***

***result=-1;***

***return result;***

***}***

**WRITING THE CODE FOR BUTTON Button.png IN EditProductFrame**

***private void btnSaveChangesActionPerformed(java.awt.event.ActionEvent evt) {***

***int result=validateInput();***

***if(result==0)***

***{***

***JOptionPane.showMessageDialog(null, "Please input numeric value for product price","Invalid Input!",JOptionPane.ERROR\_MESSAGE);***

***return;***

***}***

***if(result==-1)***

***{***

***JOptionPane.showMessageDialog(null, "Please input product name","Invalid Input!",JOptionPane.ERROR\_MESSAGE);***

***return;***

***}***

***try***

***{***

***p.setCatId(catId);***

***p.setProdId(prodId);***

***p.setProdName(prodName);***

***p.setIsActive(prodIsActive);***

***p.setProdPrice(prodPrice);***

***boolean status=ProductDao.updateProduct(p);***

***if(status)***

***{***

***JOptionPane.showMessageDialog(null, "Product successfully updated!","Success!",JOptionPane.INFORMATION\_MESSAGE);***

***}***

***else***

***JOptionPane.showMessageDialog(null, "Product not updated!","Updation Denied!",JOptionPane.ERROR\_MESSAGE);***

***}***

***catch(SQLException ex)***

***{***

***JOptionPane.showMessageDialog(null, "Error while adding product!","Exception!",JOptionPane.ERROR\_MESSAGE);***

***ex.printStackTrace();***

***}***

***}***

**WRITING THE CODE FOR BUTTON Button.png IN EditProductFrame**

***private void btnBackActionPerformed(java.awt.event.ActionEvent evt) {***

***AdminOptionsFrame adminFrame=new AdminOptionsFrame();***

***adminFrame.setVisible(true);***

***this.dispose();***

***}***