**Solution - Grails Product Information Assignment**

1. Create a domain class called "Product" with the following properties:

class Product {

Long id

String name

BigDecimal price

String description

static constraints = {

name(blank: false)

price(blank: false)

}

static mapping = {

id generator: 'assigned'

}

}

1. Create a controller called "ProductController" with the following actions:

class ProductController {

def index() {

def products = Product.list()

[products: products]

}

def create() {

[product: new Product()]

}

def save(Product product) {

if (product.validate()) {

product.save()

redirect(action: "show", id: product.id)

} else {

render(view: "create", model: [product: product])

}

}

def show(Long id) {

def product = Product.get(id)

[product: product]

}

}

1. Create a view called "create.gsp" with a form for creating a new product:

<%@ page contentType="text/html;charset=UTF-8" %>

<html>

<head>

<meta name="layout" content="main">

<title>Create Product</title>

</head>

<body>

<h1>Create Product</h1>

<g:form action="save">

<fieldset>

<div class="fieldcontain required">

<label for="name">Name:</label>

<g:textField name="name" value="${product?.name}"/>

</div>

<div class="fieldcontain required">

<label for="price">Price:</label>

<g:textField name="price" value="${product?.price}"/>

</div>

<div class="fieldcontain">

<label for="description">Description:</label>

<g:textArea name="description" value="${product?.description}"/>

</div>

</fieldset>

<div class="buttons">

<g:submitButton name="create" value="Create"/>

</div>

</g:form>

</body>

</html>

1. Validate that the name and price fields are not empty when creating a new product. This is done using the **constraints** block in the **Product** class definition.
2. Use a MySQL database to store and retrieve product information. Modify the **dataSource** block in the **grails-app/conf/application.yml** file to use your MySQL database:

dataSource:

pooled: true

jmxExport: true

driverClassName: com.mysql.jdbc.Driver

dialect: org.hibernate.dialect.MySQL5InnoDBDialect

username: your\_username

password: your\_password

url: jdbc:mysql://localhost:3306/product\_app\_db?useSSL=false

1. Create a view called "show.gsp" to display details about a specific product:

<%@ page contentType="text/html;charset=UTF-8" %>

<html>

<head>

<meta name="layout" content="main">

<title>Product Details</title>

</head>

<body>

<h1>Product Details</h1>

<table>

<tr>

<td>Name:</td>

<td>${product.name}</td>

</tr>

<tr>

<td>Price:</td>

<td>${product.price}</td>

</tr>

<tr>

<td>Description:</td>

<td>${product.description}</td>

</tr>

</table>

<g:link class="edit" action="edit" id="${product.id}">Edit</g:link>

<g:link class="delete" action="delete" id="${product.id}">Delete</g:link>

<g:link class="list" action="index">Back to List</g:link>

</body>

</html>

1. Test your application to ensure that it works correctly. Run the following command from the terminal:

grails run-app

Open a web browser and go to <http://localhost:8080/product/index>. You should see a list of products (currently empty). Click on the "Create New Product" link to create a new product.