

# **User Guide**

*VI.1*

## **Bikes and Parts Management System (BPMS)**

Bhuvesh Gupta

UID: 117326611

bgupta1@umd.edu

# Contents

<i>Requirements.....</i>	<i>3</i>
<i>Packages Required.....</i>	<i>3</i>
<i>Setting Up the Project .....</i>	<i>4</i>
<i>About the Project .....</i>	<i>8</i>
<i>User Roles and Permissions.....</i>	<i>8</i>
<i>Project Functionality Detailed .....</i>	<i>8</i>
<i>Test Account Credentials:.....</i>	<i>12</i>
<i>Test Cases .....</i>	<i>13</i>

# Requirements

1. Windows Machine
2. Microsoft Visual Studio Code 2022
3. Web Browser (Microsoft Edge, Google Chrome, Mozilla Firefox etc.)
4. ASP.NET Core 6.0 LTS
5. Git Bash

# Packages Required

1. Microsoft.AspNetCore.Identity.EntityFrameworkCore
2. Microsoft.EntityFrameworkCore.SqlServer
3. Microsoft.EntityFrameworkCore.Tools
4. Microsoft.VisualStudio.Web.CodeGeneration.Design
5. NReco.Logging.File by Vitalii Fedorchenko

# Setting Up the Project

1. Start by cloning the repository to the local machine from code.umd.edu using Git Bash at your desired location:

```
git clone git@code.umd.edu:bgupta1/ENPM809WFall2022Project-bgupta1.git
```

2. Once cloned, a folder will be created with the name ENPM809WFall2022Project-bgupta1. Navigate into the directory in Git Bash and then checkout to Phase2 branch using:

```
cd ENPM809WFall2022Project-bgupta1/ && git checkout Phase2
```

3. In Visual Studio, click on **Open a Project or Solution** and then navigate to the project folder. Then, select the **BPMS-2.sln** file which loads the project into Visual Studio.

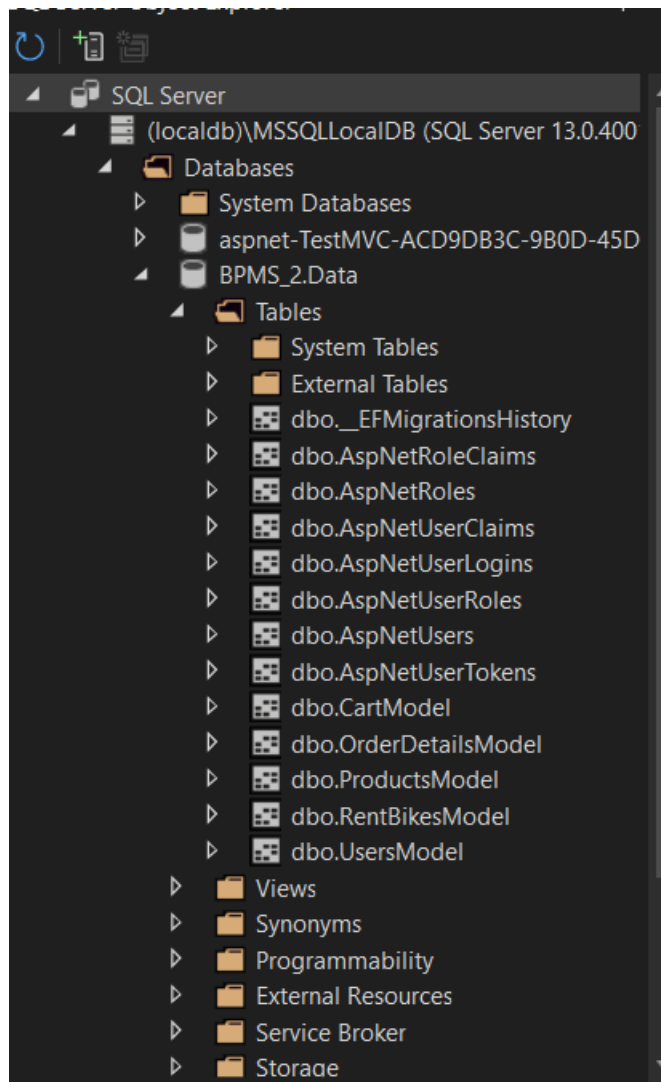
4. On initial run, navigate to **Tools ➔ NuGet Package Manager ➔ Package Manager Console**

Run the following command

```
Update-Database
```

5. To view databases, click on **View ➔ SQL Server Object Explorer**. You can find

**BPMS\_2.Data** database which is used in the web application.

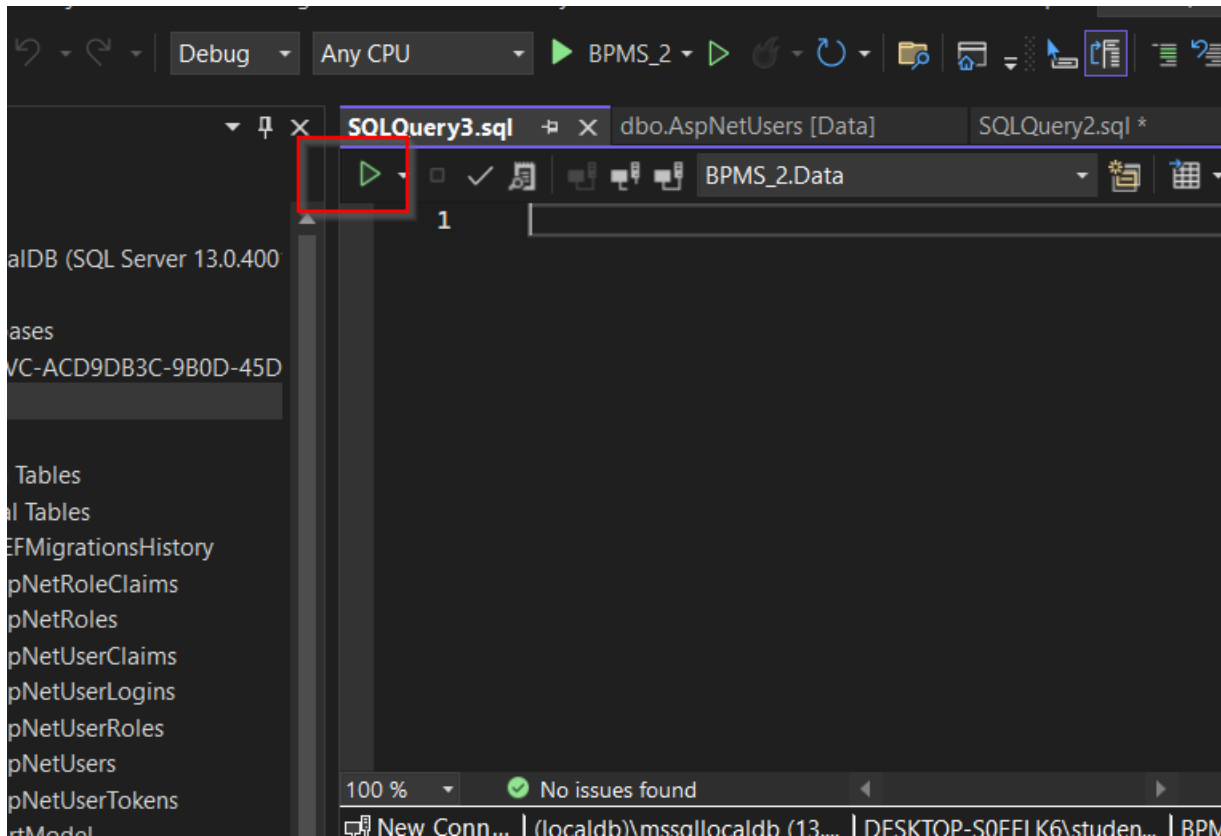


### *Setting up the Data for the Project Database*

6. After running the command right click on **BPMS\_2.Data** database and select **New Query** which opens SQL editor which will be used to run a SQL script.

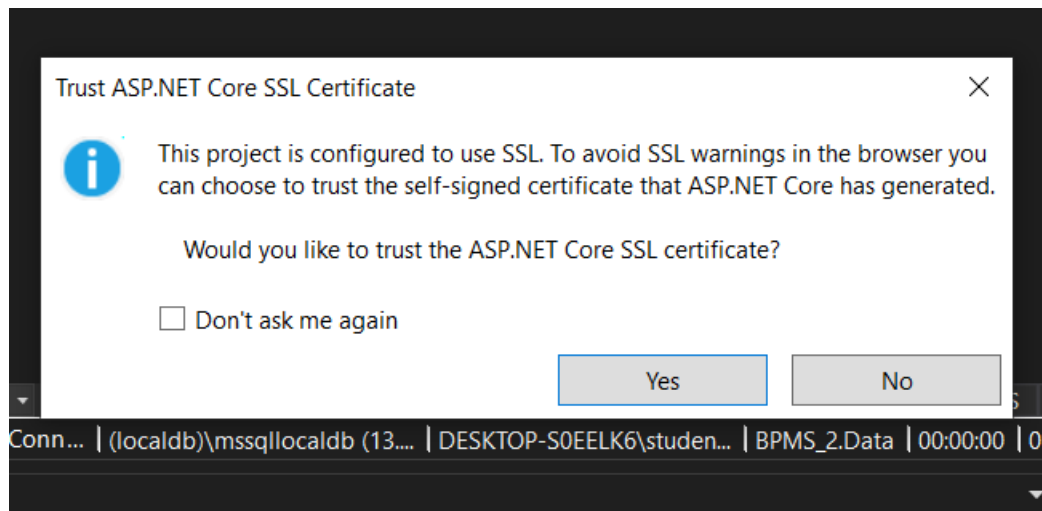
7. Open the **scripts** folder inside the directory **ENPM809WFall2022Project-bgupta1/BPMS-2/BPMS-2/** and open the file named **BPMS-2Data\_Script.sql** using Notepad or any Text Editor.

8. Copy the script and paste in the SQL Editor and execute the script using **Ctrl + Shift + E** or the green button at the top left corner.

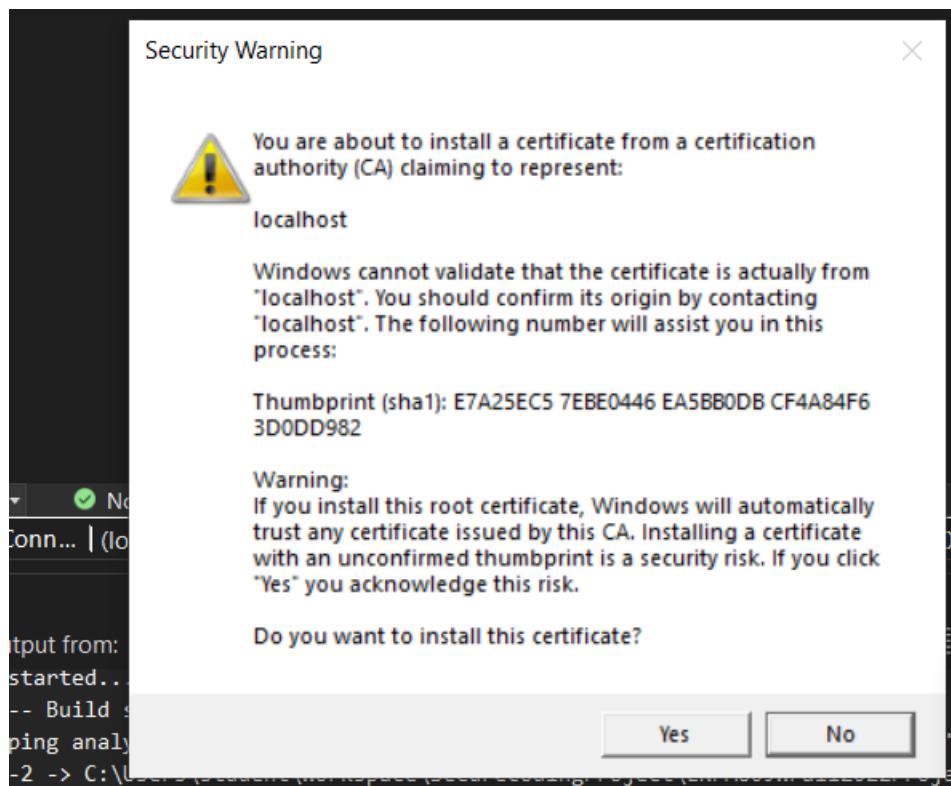


9. Now run the project by clicking on **BPMS\_2** button at the center of the toolbar which starts building the project and opens the web application in a web browser.

**\*\*If the following message pops up, then select Yes**



**\*\*Select Yes again**



# About the Project

The project is about the bikes and parts management system where the users can **buy, rent** and **return** the rented bikes. Admins **maintain the inventory** records.

## User Roles and Permissions

There are 2 user roles in the application:

**Patron** – Normal user which refers to all the bike shop customer and they have lesser access.

**Admin** – Admins with higher level of access.

## Project Functionality Detailed

### User Functions

#### ➔Register

Every student can register for an account with their umd.edu email address.

Upon clicking Register, an email would be sent to the student to verify and activate their account. They can not login without the verification.



*\*\*For the project to send registration email, please add your outlook email address at the placeholder <from\_email> and passwords at the placeholder <password> in the file ENPM809WFall2022Project-bguptal\BPMS-2\BPMS-2\Utils\EmailSender.cs*

```
SmtpClient client = new SmtpClient("smtp.office365.com")
{
    Port = 587,
    EnableSsl = true,
    UseDefaultCredentials = false,
    Credentials = new System.Net.NetworkCredential("<from_email>", "<password>", "outlook.com"),
    DeliveryMethod = SmtpDeliveryMethod.Network
};

MailMessage mailMessage = new MailMessage();
mailMessage.To.Add(new MailAddress(userEmail));
mailMessage.From = new MailAddress("<from_email>");
mailMessage.Subject = "Confirm your email";
mailMessage.Body = confirmationLink;
mailMessage.IsBodyHtml = true;

try
{

```

## ➔Login

Students can login to their verified account.

## ➔Home

Takes to the bike shop home page.

## ➔Rent Bikes

The page shows a list of bikes which can be rented by the student. A bike can be rented for 6 months only and would show the return date to the student once they rent the bike.

Only 1 bike can be rented at a time.

### ***-Keep Shopping***

After selecting the bike, the user can click this button to keep browsing more bikes before placing an order. Only 1 bike would be allowed to rent at a time. The user needs to return the previously rented bike to rent a new bike.

### ***-Checkout***

Finalizes and executed the order.

### ***-Remove From Cart***

Removes the product from the cart and lets user update its product choice.

### **➔Buy Parts and Bikes**

The page shows the various bikes and bike parts for purchase. Any number of bikes or parts can be purchased by student. Every product has its inventory count.

**\*Quantity-** It is the field where the user can enter the number of parts or bikes it needs to purchase.

### ***-Keep Shopping***

After selecting the bike or part, the user can click this button to keep browsing more bikes and parts before placing an order. The user can buy as many products as it wants but the number should be less than the inventory count.

### ***-Checkout***

Finalizes and executed the order.

### ***-Remove From Cart***

Removes the product from the cart and lets user update its product choice and quantity.

### **➔Order History**

It shows the user's history of its purchases along with the return date of its rented bike (if any)

### **➔Return Bike**

The page allows the user to return the previously rented bike. If the user exceeds the due date in returning the bike then a fine amount would be calculated for additional payment.

### **➔ Change Password**

It allows the user to change their password.

### ➔Logout

Logs out of the account.

## Admin Functions

All the functionalities which a patron has plus some additional and elevated privilege such as:

### ➔Manage Products

The admins can manage all the products by **Create, Edit** and **Delete** feature.

### ➔Manage Rental Bikes

The admins can manage the rental bikes inventory by having feature like **Create, Update** and **Delete**.

### ➔Sales

The admins can view all the orders placed by all the users and look at their total sales for financial records.

### **\*\*Assumptions:**

*The patron completes its rental order and purchases separately and does not mix the two types of purchases.*

# Test Account Credentials:

## Patron User Role

**Username:** user1

**Email:** user1@umd.edu

**Password:** VK#RRJ9GoqgD0rkZ

**Username:** user2

**Email:** user2@umd.edu

**Password:** c65sxsIIN7DR&HhZ

**Username:** user3

**Email:** user3@umd.edu

**Password:** g679SW9QNsd!l#gW

## Admin User Role

**Username:** admin1

**Email:** admin1@umd.edu

**Password:** @#oNVbk8azPX3JbJ

# Test Cases

Test Name	Unsuccessful Register Attempt
Pre Condition	<ul style="list-style-type: none"><li>• The user should be on the Home page of the application and click the Register button.</li><li>• The user entered a pre existing username.</li></ul>
Test Cases	The user enters its username, email and password and clicks on Register. It gets an error message saying “<Username> is already taken” which implies that the username already exists in the database and the user would have to choose a new one.

Test Name	Unsuccessful Register Attempt
Pre Condition	<ul style="list-style-type: none"><li>• The user should be on the Home page of the application and click the Register button.</li><li>• The user entered an email address with a domain other than umd.edu</li></ul>
Test Cases	The user enters its username, email and password and clicks on Register. It gets an error message asking the user to use its umd.edu email address which implies that the email entered by the user was not of umd.edu domain and the user would have to use its student account to register.

Test Name	Unsuccessful Register Attempt
Pre Condition	<ul style="list-style-type: none"><li>• The user should be on the Home page of the application and click the Register button.</li><li>• The user entered a weak password.</li></ul>
Test Cases	The user enters its username, email and password and clicks on Register. It gets an error message asking the user to enter the password again which complies

	with the password policies of the application. It implies that the user entered a weak password and it would have to use a stronger password.
--	---

<b>Test Name</b>	<b>Unsuccessful Register Attempt</b>
<b>Pre Condition</b>	<ul style="list-style-type: none"> <li>The user should be on the Home page of the application and click the Register button.</li> <li>The user entered different password in Confirm password field.</li> </ul>
<b>Test Cases</b>	The user enters its username, email and password and clicks on Register. It gets an error message asking the user to enter the password again because the passwords didn't match. It implies that the user entered different passwords in password and confirm password field.

<b>Test Name</b>	<b>Successful Register Attempt</b>
<b>Pre Condition</b>	<ul style="list-style-type: none"> <li>The user should be on the Home page of the application and click the Register button</li> <li>The user entered all its details correctly complying the policies.</li> </ul>
<b>Test Cases</b>	The user enters its username, email and password and clicks on Register and gets a message saying to confirm the email. It implies that the user account has been created and a verification link has been mailed to the user on its email address provided during registration.

<b>Test Name</b>	<b>Unsuccessful Login Attempt</b>
<b>Pre Condition</b>	<ul style="list-style-type: none"> <li>The user should be on the Home page of the application and click the Login button</li> <li>The user did not verify its email.</li> </ul>

<b>Test Cases</b>	The user enters its username and password and clicks on Login and gets a message saying to confirm the email. It implies that the user account has been created and a verification link has been mailed to the user on its email address provided during registration but the user has not verified its account yet.
-------------------	--

<b>Test Name</b>	<b>Unsuccessful Login Attempt</b>
<b>Pre Condition</b>	<ul style="list-style-type: none"> <li>• The user should be on the Home page of the application and click the Login button</li> <li>• The user verified its email but entered wrong credentials.</li> </ul>
<b>Test Cases</b>	The user enters its username and password and clicks on Login and gets a message saying Invalid credentials. It implies that the user either entered a wrong username or incorrect password.

<b>Test Name</b>	<b>Successful Login Attempt</b>
<b>Pre Condition</b>	<ul style="list-style-type: none"> <li>• The user should be on the Home page of the application and click the Login button</li> <li>• The user verified its email and entered correct credentials.</li> </ul>
<b>Test Cases</b>	The user enters its username and password and clicks on Login and is successfully able to log in. It implies that the user has verified its email and entered correct credentials to log in.

<b>Test Name</b>	<b>Unsuccessful Change Password Attempt</b>
<b>Pre Condition</b>	<ul style="list-style-type: none"> <li>• The user should be logged in to the application and click the Change My Password</li> <li>• The user enters incorrect old password.</li> </ul>

<b>Test Cases</b>	The user gets an error message saying that the old password is incorrect which implies that the user entered an incorrect password which it is currently using.
-------------------	---

<b>Test Name</b>	<b>Unsuccessful Change Password Attempt</b>
<b>Pre Condition</b>	<ul style="list-style-type: none"> <li>The user should be logged in to the application and click the Change My Password</li> <li>The user enters different password in confirm password.</li> </ul>
<b>Test Cases</b>	The user gets an error message saying that the passwords do not match which implies that the user entered different passwords in the new password and confirm password field.

<b>Test Name</b>	<b>Successful Change Password Attempt</b>
<b>Pre Condition</b>	<ul style="list-style-type: none"> <li>The user should be logged in to the application and click the Change My Password</li> <li>The user enters all the details correctly</li> </ul>
<b>Test Cases</b>	The user gets a success message saying that the password has been changed which implies that the password for that user has been changed and it can now use the new password to login. The old password would not work anymore.

<b>Test Name</b>	<b>No entry in Return Bike</b>
<b>Pre Condition</b>	<ul style="list-style-type: none"> <li>The user should be logged in to the application and click the Return Bike</li> <li>The user did not rent any bike or does not have any pending rentals.</li> </ul>
<b>Test Cases</b>	The user does not see anything on the page which implies that there are no pending rentals or the user did not rent anything.



Test Name	Data in Return Bike
<b>Pre Condition</b>	<ul style="list-style-type: none"> <li>The user should be logged in to the application and click the Return Bike</li> <li>The user rented a bike already.</li> </ul>
<b>Test Cases</b>	The user can see its pending rental return and can click on the return button to process the bike return. If the user exceeded its return date then a page would display the fine to the user or else there would be no fine imposed.

Test Name	User Idle
<b>Pre Condition</b>	<ul style="list-style-type: none"> <li>The user should be logged in to the application and do not perform any action for some time.</li> </ul>
<b>Test Cases</b>	The user would get logged out of the application if it is idle for a certain amount of time. The authentication cookie would be deleted and abandoned which would prevent it from being reused.

Test Name	Successful Logout
<b>Pre Condition</b>	<ul style="list-style-type: none"> <li>The user should be logged in to the application.</li> <li>The user clicks the logout button</li> </ul>
<b>Test Cases</b>	<p>The user would get logged out of the application.</p> <p>The authentication cookie would be deleted and abandoned which would prevent it from being reused.</p>

Test Name	Order History
<b>Pre Condition</b>	<ul style="list-style-type: none"> <li>The user should be logged in to the application and clicks the Order History button.</li> </ul>

<b>Test Cases</b>	The user would be able to see records of all its previous orders.
-------------------	---

<b>Test Name</b>	<b>Unsuccessful Bike Renting</b>
<b>Pre Condition</b>	<ul style="list-style-type: none"> <li>• The user should be logged in to the application and clicks Rent Bike button.</li> <li>• The user already has a pending rental to return.</li> </ul>
<b>Test Cases</b>	The user selects the bike which it wants to rent and clicks on the Checkout button. It would get a message which asks the user return the previous rental first.

<b>Test Name</b>	<b>Unsuccessful Bike Renting</b>
<b>Pre Condition</b>	<ul style="list-style-type: none"> <li>• The user should be logged in to the application and clicks Rent Bike button.</li> <li>• The user tries rent a bike which is out of stock (inventory count is 0)</li> </ul>
<b>Test Cases</b>	The user selects the bike which it wants to rent but the inventory count is 0 for that bike. It would get a message saying that the order amount can not be more that inventory count which implies that the bike is out of stock and can not be rented.

<b>Test Name</b>	<b>Unsuccessful Bike Renting</b>
<b>Pre Condition</b>	<ul style="list-style-type: none"> <li>• The user should be logged in to the application and clicks Rent Bike button.</li> <li>• The user already added another bike in cart after clicking Keep Shopping.</li> </ul>

<b>Test Cases</b>	The user selects multiple bikes which it wants to rent and clicks on the Checkout button. It would get a message which tells the user that it can only rent 1 bike at a time.
-------------------	---

<b>Test Name</b>	<b>Successful Bike Renting</b>
<b>Pre Condition</b>	<ul style="list-style-type: none"> <li>• The user should be logged in to the application and clicks Rent Bike button.</li> <li>• The user selected the bike which it wants to rent and does not have any pending rentals to return.</li> </ul>
<b>Test Cases</b>	The user is successfully able to rent the bike. Order History button would show the return date of the bike too.

<b>Test Name</b>	<b>Unsuccessful Bike/Part Purchase</b>
<b>Pre Condition</b>	<ul style="list-style-type: none"> <li>• The user should be logged in to the application and clicks Buy Parts and Bikes button.</li> <li>• The user selected the bike/part which it wants to buy.</li> <li>• The user entered the quantity more than the Inventory Count</li> </ul>
<b>Test Cases</b>	The user is gets a message saying that the quantity has to be less than the Inventory Count which implies that the user tried to buy a product whose Inventory Count is lesser than the desired quantity which the user wanted or the item is out of stock.

<b>Test Name</b>	<b>Successful Bike/Part Purchase</b>
<b>Pre Condition</b>	<ul style="list-style-type: none"> <li>• The user should be logged in to the application and clicks Buy Parts and Bikes button.</li> </ul>

	<ul style="list-style-type: none"> <li>• The user selected the bike/part which it wants to buy.</li> <li>• The user entered quantity less than the Inventory Count for all the items.</li> </ul>
<b>Test Cases</b>	The user is successfully able to buy the product. Order History button would show the details too.

### **Additional Admin Features**

<b>Test Name</b>	<b>Manage Rental Bikes Unsuccessful Update</b>
<b>Pre Condition</b>	<ul style="list-style-type: none"> <li>• The user should be logged in as admin to the application and clicks Manage Rental Bikes button</li> <li>• The user tries to Create or Edit a product and leaves a required field blank.</li> </ul>
<b>Test Cases</b>	<p>The user is not able to create or edit the product as it needs to fill in all the required fields.</p> <p>The user can delete the products.</p>

<b>Test Name</b>	<b>Manage Rental Bikes Successful Update</b>
<b>Pre Condition</b>	<ul style="list-style-type: none"> <li>• The user should be logged in as admin to the application and clicks Manage Rental Bikes button</li> <li>• The user tries to Create or Edit a product and enters all the details correctly.</li> </ul>
<b>Test Cases</b>	<p>The user is able to create or edit the product as it needs to fill in all the required fields.</p> <p>The user can delete the products too.</p>

<b>Test Name</b>	<b>Manage Bikes and Parts Unsuccessful Update</b>
<b>Pre Condition</b>	<ul style="list-style-type: none"> <li>• The user should be logged in as admin to the application and clicks Manage Bikes and Parts button</li> <li>• The user tries to Create or Edit a product and leaves a required field blank.</li> </ul>
<b>Test Cases</b>	<p>The user is not able to create or edit the product as it needs to fill in all the required fields.</p> <p>The user can delete the products.</p>

<b>Test Name</b>	<b>Manage Bikes and Parts Successful Update</b>
<b>Pre Condition</b>	<ul style="list-style-type: none"> <li>• The user should be logged in as admin to the application and clicks Manage Bikes and Parts button</li> <li>• The user tries to Create or Edit a product and enters all the details correctly.</li> </ul>
<b>Test Cases</b>	<p>The user is able to create or edit the product as it needs to fill in all the required fields.</p> <p>The user can delete the products too.</p>

<b>Test Name</b>	<b>Sales Page</b>
<b>Pre Condition</b>	<ul style="list-style-type: none"> <li>• The user should be logged in as admin to the application and clicks Sales button</li> </ul>
<b>Test Cases</b>	The user is able to see all the sales of the shop.