Retail Store Stock Inventory Analytics

1.INTRODUCTION

1.1 Project Overview:

Project is based on Retail Store Stock Inventory analytics which is used to supply the stocks for shops based on their needs .By managing inventory, retailers meet customer demand without running out of stock or carrying excess supply.

Inventory management is vital for retailers because the practice helps them increase profits. They are more likely to have enough inventory to capture every possible sale while avoiding overstock and minimizing expenses.

1.2 Purpose:

Purpose of retail store stock analysis is to find the necessary stock required for to supply customer when there are in need of , the shop holder view the stock,price and sale in form dashboard,report and story in webpage which helps them to track regularly the status of their stock availability.

It helps in managing the current stock levels, ordered items and products as well as ones already sold. It provides a constant supply of products to fulfill customer demand. It allows customer retention. Customers convert into loyal customers by handling stock levels.

2.LITERATURE SURVEY

2.1 Existing problem:

Existing system consist of methods using ABC Analysis and Min-Max Analysis. In the Data Mart, the search for goods classes per sub category is carried out using the ABC Analysis calculation method. Furthermore, in the Data Mart, the search for maximum and minimum stock values is based on the Min-Max Analysis calculation method. The resulting maximum and minimum grade and stock values are then implemented into the goods

data table in the retail management information system database. The last stage is to arrange the order amount that is allowed in the order module in the retail management information system. Rules that are made based on the class of goods along with the minimum and maximum stock values.

2.2 References:

The following are the references used:

[1] H S Sugiarto and H T Saksono 2016 Scheduling System on Goods Order At PT XYZ Using Economic Order Quantity Method The Third International Conference on Entrepreneurship.

[2] K E Fu and P Apichotwasurat 2013 Application of Economic Order Quantity on Production Scheduling and Control System for a Small Company. Proceedings of the Institute of Industrial Engineers Asian Conference 2013.

 [3] M Rusănescu 2014 Abc Analysis , Model for Classifying Inventory HIDRAULICA.

 [4] D Dhoka and Y L Choudary 2013 ABC Classification for Inventory Optimization IOSR J Bus Manage.

[5] Funaki, K.., "Strategies safety stock placement in supply chain design with due-date based demand," International Journal of Production Economics, vol. 135, pp 4-13, 2012. [6] Grewal, CS, Enns, ST, and Rogers, P., "Dynamic reorder point replenishment strategies for a capacitated supply chain with seasonal demand," Computer, and industrial engineering, vol. .80, pp 97-110, 2015.

[7] Indrajit, RE, and Djokopranoto, R., "General merchandise and inventory management of spare parts for maintenance, repair and operation", Yogyakarta: Grasindo, 2014.

 [8] Mebarki, N. and Shahzad, A., "Correlation among tardiness based

measures for priority scheduling using dispatching rules" Month, pp 1- 14, 2012.

## 2.3 Problem Statement Definition:

The shop holder sell the product as requested by the consumer at the time when the stock is not available it will reduce customer satisfaction and it is also a loss for the seller.

In order to overcome this drawback we using visualization to represent stocks availability and sale.Therefore quality and quantity of product can be delivered without any demand.

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	A Shopaholic	Buy my favourite products	Most of the products are not available in store	But the demand for the product is high	Unsatisfied
PS-2	A Foodie	Buy my favourite chocolate	It is not available in any stores	It is imported chocolates	Frustrated

Problem Statement-1:

I am	I'm trying to	But	Because	Which makes me feel
A shopaholic	Buy my favourite products	Most of the products are not available in the store	But the demand for the product is high	Unsatisfied

Problem Statement-2:

I am	I'm trying to	But	Because	Which makes me feel
A foodie	Buy my favourite chocolates	It is not available in any stores	It is imported chocolates	Frustrated
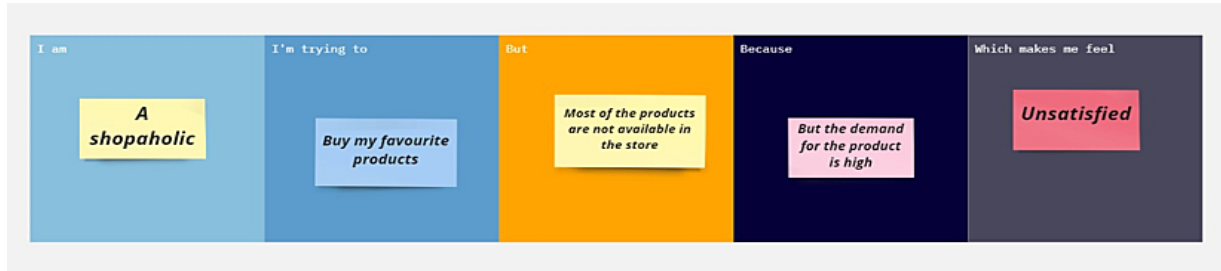
3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas:

An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. Much like a user persona, an empathy map can represent a group of users, such as a customer segment.

An Empathy Map consists of four quadrants. The four quadrants reflect four key traits, which the user demonstrated/possessed during the observation/research stage. The four quadrants refer to what the user:

Retail Store Stock Inventory Analytics

Said, Did, Thought, and Felt. It's fairly easy to determine what the user said and did.
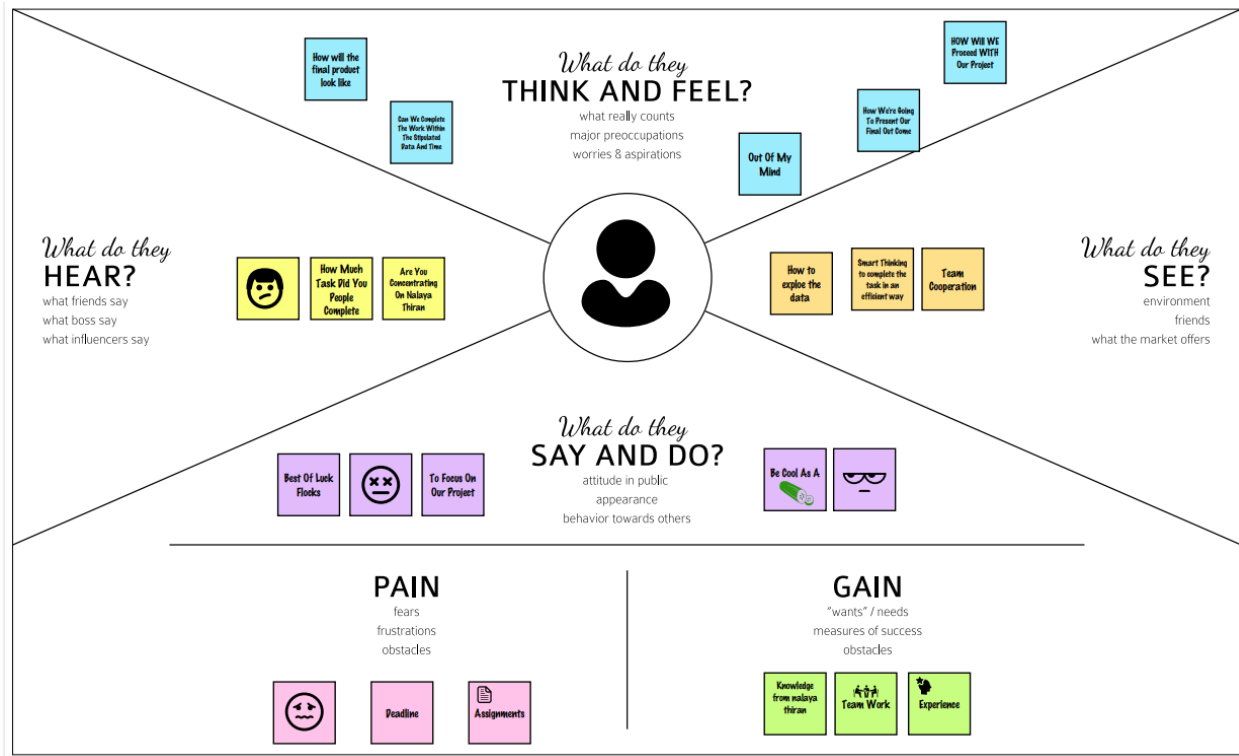
3.2 Ideation & Brainstorming:

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

Retail Store Stock Inventory Analytics

Step-1: Team Gathering, Collaboration and Select the Problem Statement

Step-2: Brainstorm, Idea Listing and Grouping

Retail Store Stock Inventory Analytics

Step-3: Idea Prioritization
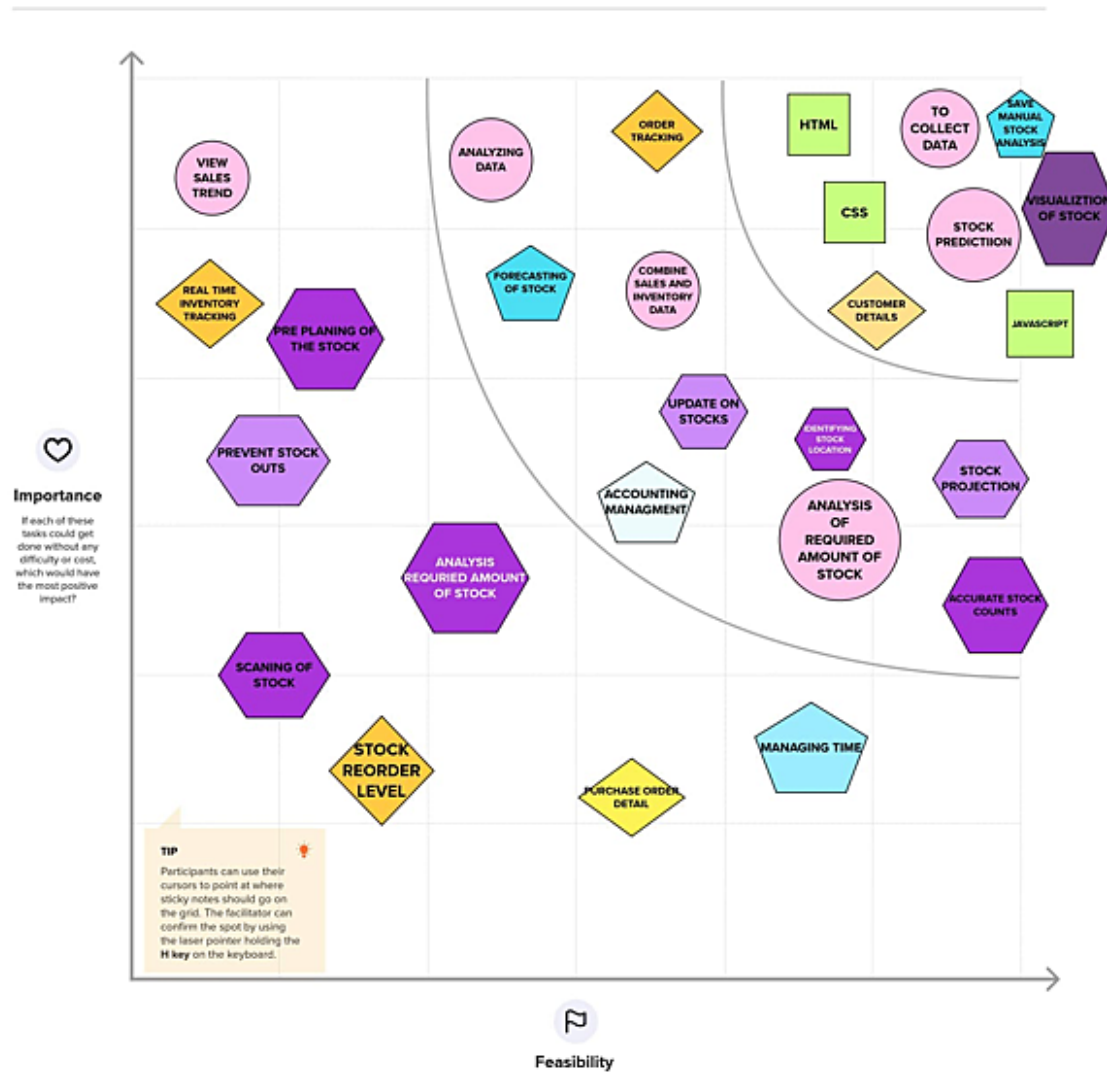
Retail Store Stock Inventory Analytics

3.3 Proposed Solution:

Proposed Solution means the technical solution to be provided by the Implementation agency in response to the requirements and the objectives of the Project.

The main goal of presenting a business proposal is to provide solution to a problem faced by a potential buyer. This section should be as comprehensive as possible, and able to address all the needs that you have pointed in the first section.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Customers gets disappointed when the store does not meets the satisfaction level of them.
2.	Idea / Solution description	Using dashboard it would become easy for the store to keep a track on their stock, so that they can meet customer's satisfaction level.
3.	Novelty / Uniqueness	Expiry alert of the product will be given.
4.	Social Impact / Customer Satisfaction	Quality and Quantity of the product can maintained to the best, and customer's will have a heart full feeling while leaving the store.
5.	Business Model (Revenue Model)	Using this method the company will have reputed customers and stocks will be delivered on time, so there is no need of last minute hassle.

6.	Scalability of the Solution	When your inventory is hard to identify or locate in the warehouse, it leads to incomplete, inaccurate or delayed shipments. Receiving and finding the right stock is vital to efficient warehouse operations and provides a positive customer experiences.

3.4 Problem Solution fit:

Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem.The Problem-Solution Fit is an important step towards the Product-Market Fit, but often an underestimated one.

Problem-Solution canvas is a tool for entrepreneurs, marketers and corporate innovators, which helps them identify solutions with higher chances for solution adoption, reduce time spent on solution testing and get a better overview of current situation.

Retail Store Stock Inventory Analytics

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
Shopkeeper's who are in need for buying products to refill their stocks.

6. CUSTOMER CONSTRAINTS

What constraints prevent your customers from taking action or limit their choices of solutions? As personal computer won't be available to everyone and most of the people won't be knowing the technology that they are using, even if they have to allot a person for in charge they have to pay for them, so they are worried to go for.

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customer when they face the problem or need to get the job done? what have they tried in the past?
Customers assign a managing assistant to over view the stock

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers?
Shopkeeper will able to see the products which they need to buy based on the availability and also they get to know about the product expiry date so that they could dispose them in advance.

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists? What is the back story behind the need to do this job?

Most of the shopkeepers lose their customers because of insufficient stock and proper stock management. So they face loss in their business. In order to help them with their profit we are helping them with technology to make it easy.

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?
They will seek a software engineer and explain all the requirements to be put in the dashboard and learn in and out of the dashboard and know what profit would they make out of that and implement it.

3. TRIGGERS TR

What triggers customers to act?
We will be showing our products through dashboard which makes the shopkeeper to trigger.

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?
 The shopkeeper visits the warehouse directly and check the availability of the product where by using the dashboard there is no need to visit the warehouse rather they get view the details of the product and the availability in their dashboard itself which saves their time.

10. YOUR SOLUTION SL

We are working on a new business proposition, where the shopkeeper can view the products availability in form of dashboard and buy product based on their need which saves time for them. Expiry date is displayed which helps shopkeeper to provide quantity and quality product to customer.

8.CHANNELS of BEHAVIOUR CH

8.1 ONLINE
They learn how to make use of new technologies and how do they get profited out of it

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.
They will apply the sources in the real time whatever they learnt.

4.REQUIREMENT ANALYSIS

4.1 Functional requirement:

Functional requirements may involve calculations, technical details, data manipulation and processing, and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describe all the cases where the system uses the functional requirements, these are captured in use cases.

Functional requirements drive the application architecture of a system, while non-functional requirements drive the technical architecture of a system.

Retail Store Stock Inventory Analytics

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through webpage
FR-2	User Login	Login through webpage
FR-3	User Stock List	View in the webpage
FR-4	Sales List	View in the webpage
FR-5	Revenue Detail	View in the webpage

4.2 Non-Functional requirements:

Non-functional requirements are often mistakenly called the "quality attributes" of a system, however there is a distinction between the two. Non-functional requirements are the criteria for evaluating how a software system should perform and a software system must have certain quality attributes in order to meet non-functional requirements.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Clear view about Stock Details which provides benefit to the users.
NFR-2	**Security**	Only authorized users can log in to view that provide security to the users.
NFR-3	**Reliability**	Information about one store cannot be viewed by the other

		store users.
NFR-4	**Performance**	Available and required amount of stock can be viewed in dashboard by visualization hence the user can make decision according to it.
NFR-5	**Availability**	Visualization shows the stock availability and the products which need to be refilled can be viewed that prevent user from last minute shortage.
NFR-6	**Scalability**	Product expiry date can be viewed which helps the user to sell those product in prior that provides benefits to the shop owner.

5.PROJECT DESIGN

5.1 Data Flow Diagrams:

A data-flow diagram is a way of representing a flow of data through a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow — there are no decision rules and no loops.

FIRST LEVEL DATA FLOW DIAGRAM

5.2 Solution & Technical Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

Retail Store Stock Inventory Analytics

RETAIL STOCK INVENTORY MANAGEMENT

Technical Architecture (TA) is a form of IT architecture that is used to design computer systems. It involves the development of a technical blueprint with regard to the arrangement, interaction, and interdependence of all elements so that system-relevant requirements are met.
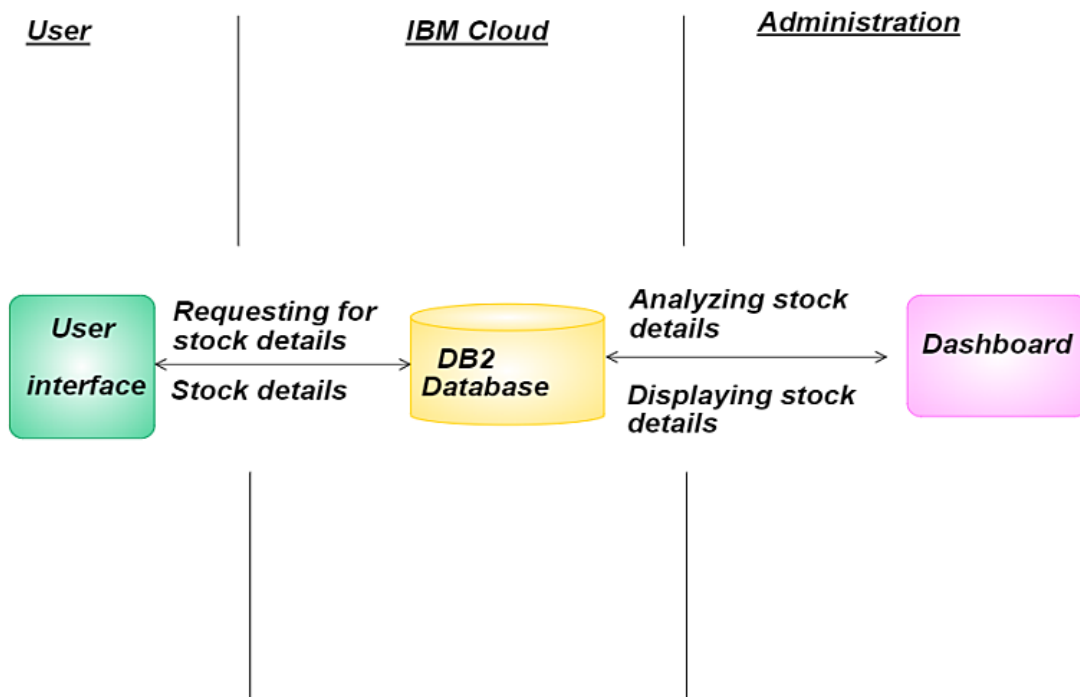
Retail Store Stock Inventory Analytics

5.3 User Stories:

A user story is an informal, general explanation of a software feature written from the perspective of the end user or customer. The purpose of a user story is to articulate how a piece of work will deliver a particular value back to the customer.

In software development and product management, a user story is an informal, natural language description of features of a software system.

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	Medium	Suchitra Ramaraj, Ragavi.S
Sprint-1	Login	USN-2	As a user, I can log into the application by entering username& password.	2	Medium	Varshetha.M, Sharon Shamitha.S
Sprint-2	Data Upload	USN-3	As a user, I can upload my data so that I can have a visual representation of it.	1	Low	Ragavi.S
Sprint-2	Dashboard Creation	USN-4	As a user, I can view the visual representation Of my data in dashboard.	3	High	Rathnaa.S, Suchitra Ramaraj
Sprint-3	Report Creation	USN-5	As a user, I can view the visual representation Of my data in report.	3	High	Varshetha.M, Sharon Shamitha.S
Sprint-4	Story Creation	USN-6	As a user, I can view the visual representation Of my data in story.	3	High	Ragavi.S, Rathnaa.S

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation:

In Scrum Projects, Estimation is done by the entire team during Sprint Planning Meeting. The objective of the Estimation would be to consider the User Stories for the Sprint by Priority and by the Ability of the team to deliver during the Time Box of the Sprint.

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	30 Oct 2022	20	30 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	06 Nov 2022	20	06 Nov 2022
Sprint-3	20	6 Days	06 Nov 2022	13 Nov 2022	20	13 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.2 Sprint Delivery Schedule:

In Scrum project sprint delivery schedule is used to estimate when sprint has started and delivery date of the sprint.Due to estimation of the sprint delivery schedule it helps the developer to complete their project within the estimated time.

6.3 Reports from JIRA:

The reports in jira has been denoted below:

BACKLOG:

Backlog is usually a list of issues describing what your team is going to do on a project. It's a convenient place for creating, storing, and managing several kinds of issues: issues that you're currently working on (you can also see them on the board and in the current sprint if you're using a Scrum project).

Retail Store Stock Inventory Analytics

BOARD:

A board displays your team's work as cards you can move between columns. In Jira Software, cards and the tasks they represent are called "issues". Usually, your board reflects your team's process, tracking the status of work as it makes its way through your team's process.

Retail Store Stock Inventory Analytics

COMMULATIVE FLOW DIAGRAM:

A Cumulative Flow Diagram (CFD) is an area chart that shows the various statuses of work items for an application, version, or sprint. The horizontal x-axis in a CFD indicates time, and the vertical y-axis indicates cards (issues).

Retail Store Stock Inventory Analytics

ROAD MAP:

Roadmaps in Jira Software are team-level roadmaps useful for planning large pieces of work several months in advance at the Epic level within a single project. Simple planning and dependency management features help your teams visualize and manage work better together.
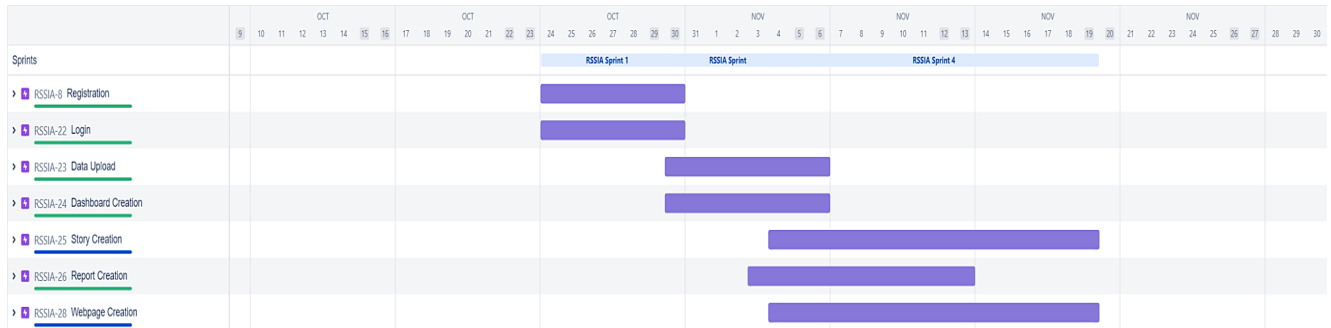
SPRINT BURNDOWN CHART:

A burndown chart shows the amount of work that has been completed in an epic or sprint, and the total work remaining. Burndown charts are used to predict your team's likelihood of completing their work in the time available.

SPRINT1:
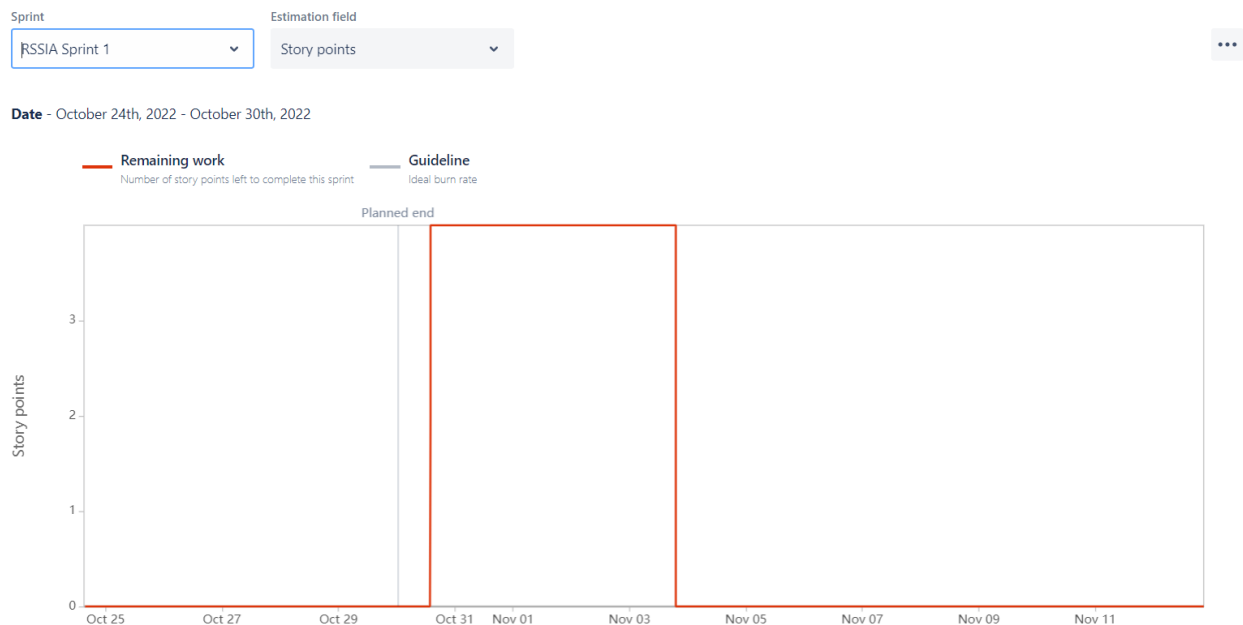
SPRINT2:

Retail Store Stock Inventory Analytics

Sprint
RSSIA Sprint 2 ▾

Estimation field
Story points ▾

···

Date - October 30th, 2022 - November 6th, 2022

— **Remaining work**
Number of story points left to complete this sprint

— **Guideline**
Ideal burn rate
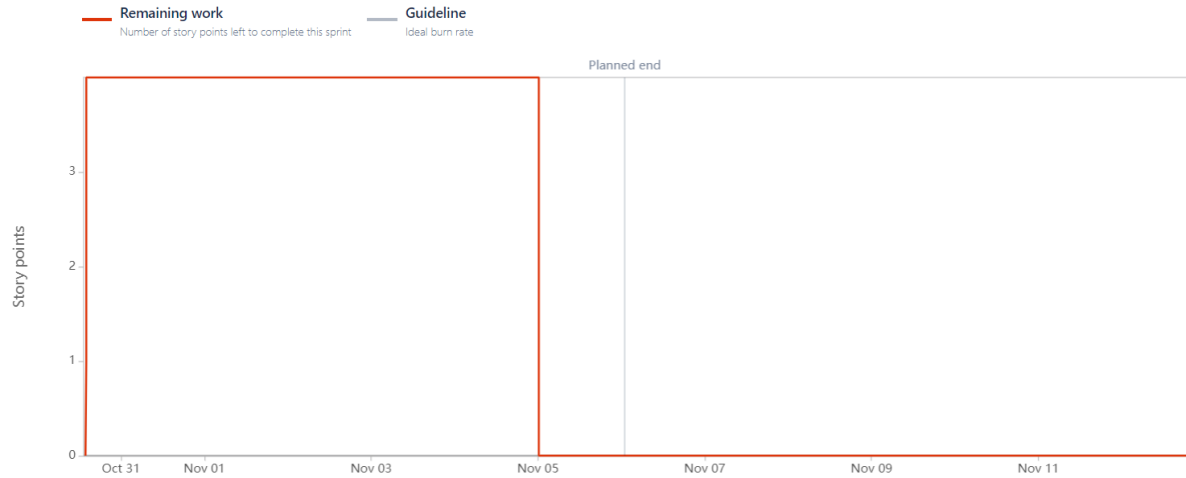
SPRINT3:

Sprint
RSSIA Sprint 3 ▾

Estimation field
Story points ▾

···

Date - November 3rd, 2022 - November 13th, 2022

— **Remaining work**
Number of story points left to complete this sprint

— **Guideline**
Ideal burn rate

SPRINT4:

Date - November 4th, 2022 - November 19th, 2022

Remaining work
Number of story points left to complete this sprint

Guideline
Ideal burn rate
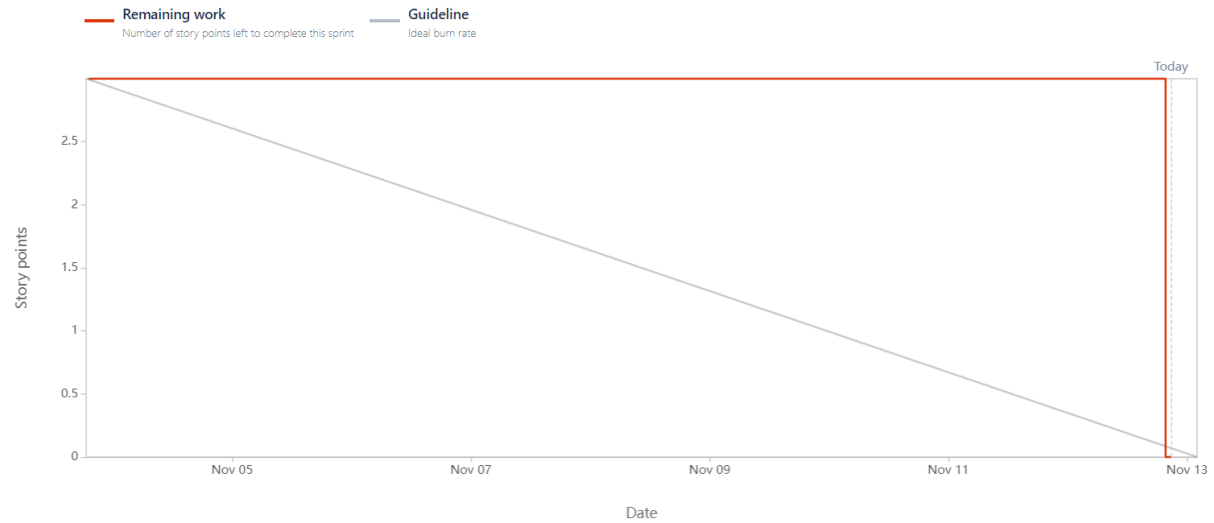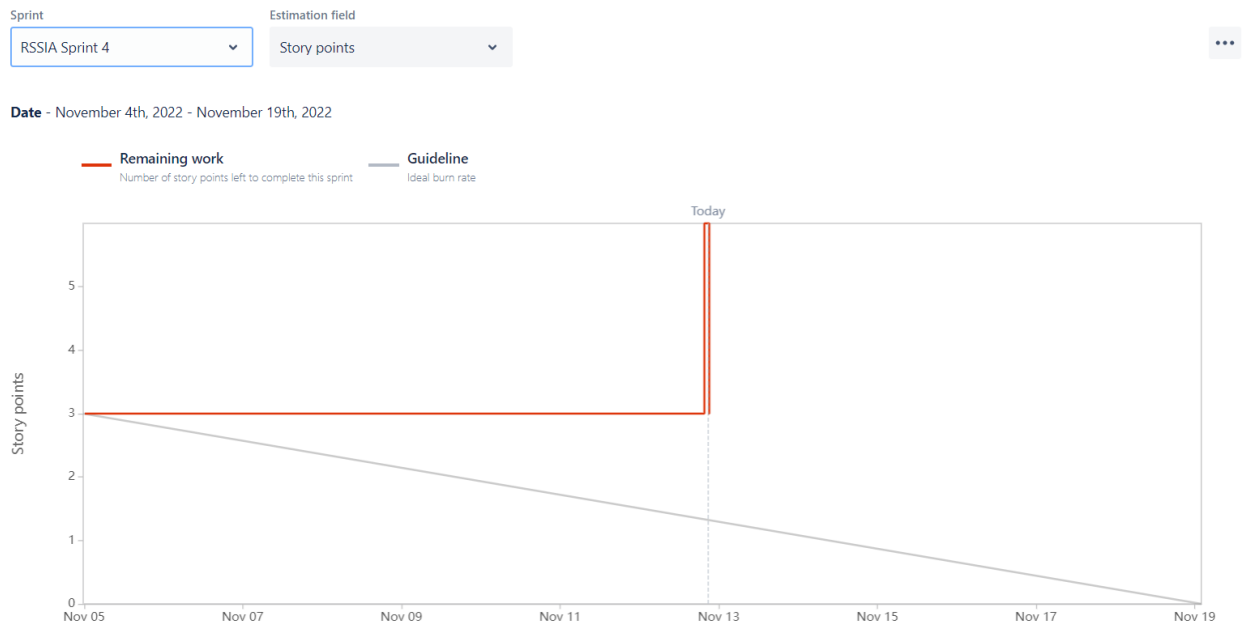
7. CODING & SOLUTIONING

7.1 Feature 1: Login

- Login page consist of two field:
- ✓ Username
- ✓ Password
- When a new user fill the username and password and click "login now" button available it shows alert message.
- When a registered user fill the username and password and click "login now" button available it redirects the user to the home page with container consisting of a welcome note along with the users name will be displayed .
- The feature which we have implemented in our login page is that, it will count the number of times the user has logged in and stores it in the database.

Code:

```php
<?php
use Phppot\Member;
```

Retail Store Stock Inventory Analytics

```php
if (! empty($_POST["login-btn"])) {
    require_once __DIR__ . '/Model/Member.php';
    $member = new Member();
    $loginResult = $member->loginMember();
}
?>
<HTML>
<HEAD>
<TITLE>Login</TITLE>
<link href="assets/css/phppot-style.css" type="text/css"
        rel="stylesheet" />
<link href="assets/css/user-registration.css" type="text/css"
        rel="stylesheet" />
<script src="vendor/jquery/jquery-3.3.1.js" type="text/javascript"></script>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" integrity="sha384-
BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68v
bdEjh4u" crossorigin="anonymous">
<style>
        body{
    background-
image:url("https://media.istockphoto.com/id/913219882/photo/financial-
graph-on-technology-abstract-
background.jpg?b=1&s=170667a&w=0&k=20&c=aFXB_2Bn_T7weuTYIqZAs
8sIja9MHcT-u-TlBVftvqo=");
        background-position: center; /* Center the image */
  background-repeat: no-repeat; /* Do not repeat the image */
  background-size: cover; /* Resize the background image to cover the
entire container */
```

```
    }
.sign-up-container{
  background-image: linear-gradient(to right, #ff0030 , #790975);
}
.form-label{
color:white !important;
}
#login-btn{
        color:white;
        font-weight:bold;
        background: #343a40;
}
</style>
</HEAD>
<BODY>
        <div class="phppot-container">
                <div class="sign-up-container">
                        <div class="login-signup">
                                <a href="user-registration.php"
style="color:white;">Sign up</a>
                        </div>
                        <div class="signup-align">
                                <form name="login" action="" method="post"
                                        onsubmit="return loginValidation()">
                                        <div class="signup-heading"
style="color:white;">Login</div>
                                        <?php if(!empty($loginResult)){?>
                                        <div class="error-msg"><?php echo
$loginResult;?></div>
                                        <?php }?>
                                        <div class="row">
```
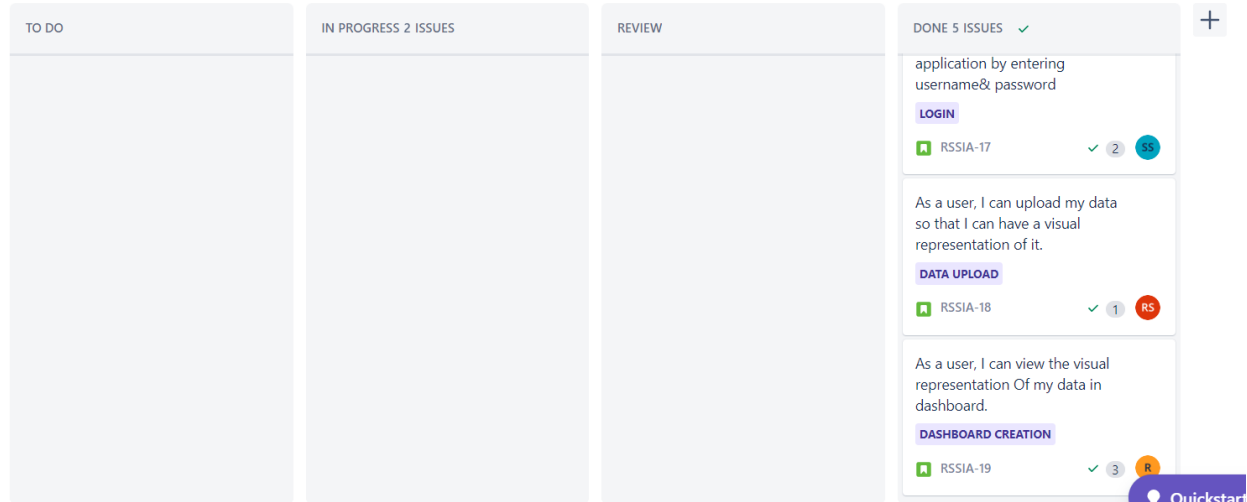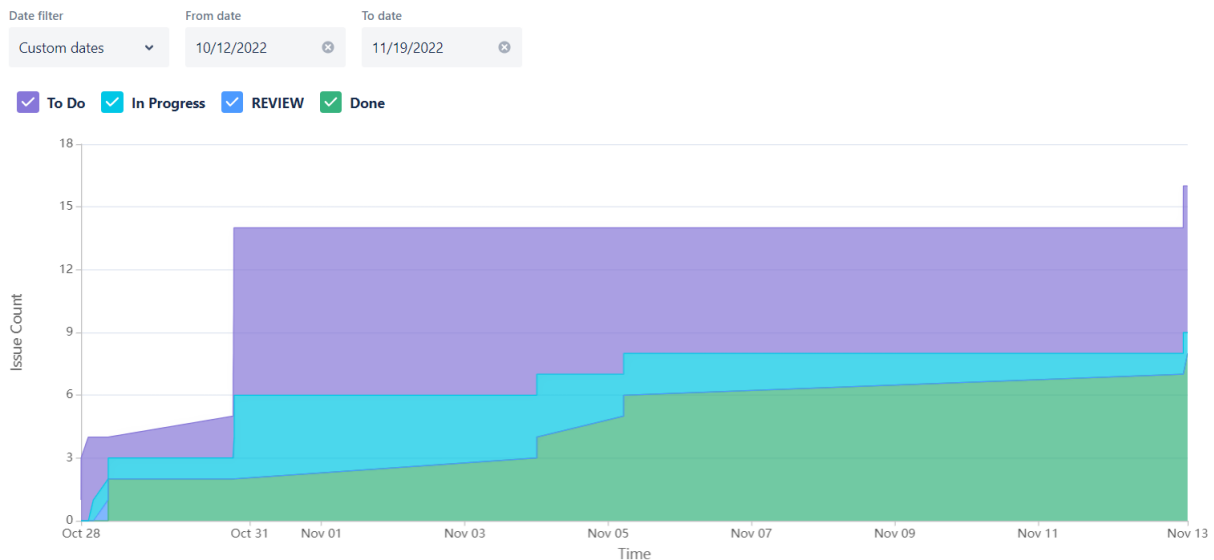
Retail Store Stock Inventory Analytics

```html
<div class="inline-block">
    <div class="form-label">
        Username<span class="required error" id="username-info"></span>
    </div>
    <input class="input-box-330" type="text" name="username" id="username">
</div>
</div>
<div class="row">
    <div class="inline-block">
        <div class="form-label">
            Password<span class="required error" id="login-password-info"></span>
        </div>
        <input class="input-box-330" type="password" name="login-password" id="login-password">
    </div>
</div>
<div class="row">
    <input class="btn btn-dark" type="submit" name="login-btn" id="login-btn" value="Login Now">
</div>
</form>
</div>
</div>
</div>
```

Retail Store Stock Inventory Analytics

```
<script>
function loginValidation() {
      var valid = true;
      $("#username").removeClass("error-field");
      $("#password").removeClass("error-field");

      var UserName = $("#username").val();
      var Password = $('#login-password').val();

      $("#username-info").html("").hide();

      if (UserName.trim() == "") {
            $("#username-info").html("required.").css("color",
"#ee0000").show();
            $("#username").addClass("error-field");
            valid = false;
      }
      if (Password.trim() == "") {
            $("#login-password-info").html("required.").css("color",
"#ee0000").show();
            $("#login-password").addClass("error-field");
            valid = false;
      }
      if (valid == false) {
            $('.error-field').first().focus();
            valid = false;
      }
      return valid;
}
</script>
```

```
</BODY>
</HTML>
```

7.2 Feature 2: <u>Registration</u>

- Registration page consist of four fields:
- ✓ Username
- ✓ Email
- ✓ Password
- ✓ Confirm Password
- When a new user fill the login and click "login now" button it show a alert message, so that the user should sign up first.
- After that the user fills the registration form and clicks the "Sign up" button the user will be registered.
- Now the user will able to login to view the home page.

Code:

```php
<?php
use Phppot\Member;
if (! empty($_POST["signup-btn"])) {
    require_once './Model/Member.php';
    $member = new Member();
    $registrationResponse = $member->registerMember();
}
?>
<HTML>
<HEAD>
<TITLE>User Registration</TITLE>
<link href="assets/css/phppot-style.css" type="text/css"
    rel="stylesheet" />
<link href="assets/css/user-registration.css" type="text/css"
    rel="stylesheet" />
<script src="vendor/jquery/jquery-3.3.1.js" type="text/javascript"></script>
```

```
</HEAD>
<style>
            body{
    background-
image:url("https://i.pinimg.com/736x/1a/f1/c6/1af1c62ecf24b3d90a9f4f2
05914e27e--business-analyst-data-analytics.jpg");
        background-position: center; /* Center the image */
  background-repeat: no-repeat; /* Do not repeat the image */
  background-size: cover; /* Resize the background image to cover the
entire container */
    }
.sign-up-container{
  background-image: linear-gradient(to right, #790975,#ff0030);
}
.form-label{
color:white !important;
}
#signup-btn{
        color:white;
        font-weight:bold;
        background: #343a40;
}
</style>
<BODY>

        <div class="phppot-container">
                <div class="sign-up-container">
                        <div class="login-signup">
                                <a href="index.php" style="color:white">Login</a>
                        </div>
                        <div class="">
```

Retail Store Stock Inventory Analytics

```php
<form name="sign-up" action="" method="post"
    onsubmit="return signupValidation()">
    <div class="signup-heading"
style="color:white">Registration</div>
    <?php
if (! empty($registrationResponse["status"])) {
    ?>
        <?php
    if ($registrationResponse["status"] == "error") {
        ?>
            <div class="server-response error-msg"><?php
echo $registrationResponse["message"]; ?></div>
        <?php
    } else if ($registrationResponse["status"] == "success") {
        ?>
            <div class="server-response success-msg"><?php echo
$registrationResponse["message"]; ?></div>
        <?php
    }
    ?>
        <?php
}
?>  <div class="error-msg" id="error-msg"></div>
    <div class="row">
        <div class="inline-block">
            <div class="form-label">
                Username<span
class="required error" id="username-info"></span>
            </div>
            <input class="input-box-330"
type="text" name="username"
```

```
                                                id="username">
                                        </div>
                                </div>
                                <div class="row">
                                        <div class="inline-block">
                                                <div class="form-label">
                                                        Email<span class="required
error" id="email-info"></span>
                                                </div>
                                                <input class="input-box-330"
type="email" name="email" id="email">
                                        </div>
                                </div>
                                <div class="row">
                                        <div class="inline-block">
                                                <div class="form-label">
                                                        Password<span
class="required error" id="signup-password-info"></span>
                                                </div>
                                                <input class="input-box-330"
type="password"
                                                        name="signup-password"
id="signup-password">
                                        </div>
                                </div>
                                <div class="row">
                                        <div class="inline-block">
                                                <div class="form-label">
                                                        Confirm Password<span
class="required error"
                                                                id="confirm-password-
```

info">

</div>

<input class="input-box-330"

type="password"

name="confirm-password"

id="confirm-password">

</div>

</div>

<div class="row">

<input class="btn" type="submit"

name="signup-btn"

id="signup-btn" value="Sign up">

</div>

</form>

</div>

</div>

</div>

<script>

```
function signupValidation() {
    var valid = true;
    $("#username").removeClass("error-field");
    $("#email").removeClass("error-field");
    $("#password").removeClass("error-field");
    $("#confirm-password").removeClass("error-field");
    var UserName = $("#username").val();
    var email = $("#email").val();
    var Password = $('#signup-password').val();
  var ConfirmPassword = $('#confirm-password').val();
    var emailRegex = /^[a-zA-Z0-9.!#$%&'*+/=?^_`{|}~-]+@[a-zA-Z0-9](?:[a-
zA-Z0-9-]{0,61}[a-zA-Z0-9])?(?:\.[a-zA-Z0-9](?:[a-zA-Z0-9-]{0,61}[a-zA-Z0-
9])?)*$/;
```

```
$("#username-info").html("").hide();
$("#email-info").html("").hide();
if (UserName.trim() == "") {
        $("#username-info").html("required.").css("color",
"#ee0000").show();
        $("#username").addClass("error-field");
        valid = false;
}
if (email == "") {
        $("#email-info").html("required").css("color", "#ee0000").show();
        $("#email").addClass("error-field");
        valid = false;
} else if (email.trim() == "") {
        $("#email-info").html("Invalid email address.").css("color",
"#ee0000").show();
        $("#email").addClass("error-field");
        valid = false;
} else if (!emailRegex.test(email)) {
        $("#email-info").html("Invalid email address.").css("color",
"#ee0000")
                        .show();
        $("#email").addClass("error-field");
        valid = false;
}
if (Password.trim() == "") {
        $("#signup-password-info").html("required.").css("color",
"#ee0000").show();
        $("#signup-password").addClass("error-field");
        valid = false;
}
if (ConfirmPassword.trim() == "") {
```

```
        $("#confirm-password-info").html("required.").css("color",
"#ee0000").show();
        $("#confirm-password").addClass("error-field");
        valid = false;
    }
    if(Password != ConfirmPassword){
   $("#error-msg").html("Both passwords must be same.").show();
   valid=false;
 }
    if (valid == false) {
        $('.error-field').first().focus();
        valid = false;
    }
    return valid;
}
</script>
</BODY>
</HTML>
```

7.3 Feature 3: <u>Display of Dashboard, Report and Story</u>

- After the user has successfully registered and login the user will be redirected to the home page with welcome note "Welcome to Account" along with the username.
- The home page consist of 4 buttons:
- ✓ Story
- ✓ Report
- ✓ Dashboard
- ✓ Logout
- Based on the user preference they can view the story,report and dashboard by clicking on the respective buttons.
- After viewing the required preferences the user can leave the

Retail Store Stock Inventory Analytics

webpage by clicking the "Logout" button available in the home page.

Code:

```php
<?php
session_start();
if (isset($_SESSION["username"])) {
    $username = $_SESSION["username"];
    session_write_close();
} else {
    // since the username is not set in session, the user is not-logged-in
    // he is trying to access this page unauthorized
    // so let's clear all session variables and redirect him to index
    session_unset();
    session_write_close();
    $url = "./index.php";
    header("Location: $url");
}

?>
<HTML>
<HEAD>
<TITLE>Welcome</TITLE>
<link href="assets/css/phppot-style.css" type="text/css"
    rel="stylesheet" />
<link href="assets/css/user-registration.css" type="text/css"
    rel="stylesheet" />
  <!-- Latest compiled and minified CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" integrity="sha384-
BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68v
bdEjh4u" crossorigin="anonymous">
```

Retail Store Stock Inventory Analytics

```
<style>
   body{
      background-image:url("https://img.freepik.com/premium-
photo/magnifying-glass-documents-with-analytics-data-lying-table_33799-
7624.jpg?w=2000");
      background-position: center; /* Center the image */
   background-repeat: no-repeat; /* Do not repeat the image */
   background-size: cover; /* Resize the background image to cover the
entire container */
   }
.dropdown {
  position: absolute;
  display: inline-block;
}

.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f1f1f1;
  min-width: 160px;
  z-index: 1;
}

.dropdown-content a {
  color: black;
  padding: 12px 16px;
  text-decoration: none;
  display: block;
}

.dropdown-content a:hover {background-color: #ddd}
```

Retail Store Stock Inventory Analytics

```html
.dropdown:hover .dropdown-content {
  display: block;
}

.btn:hover, .dropdown:hover .btn {
  background-color: #6b2139d6;
}
</style>
</HEAD>
<BODY>
    <!-- <img src="./assets/bg.jpg" alt=""> -->
        <div class="phppot-container">
    <a href = "./assets/story.html">
    <div class="col btn btn-danger" style="margin-top:45px;right: 0">
Story</div>
     </a>


    <div class="dropdown">
      <div class="col btn btn-danger" style="margin-top:45px;right: 0">
Report</div>
        </button>
     <div class="dropdown-content">
     <a href="./assets/report1.html">Report 1</a>
     <a href="./assets/report2.html">Report 2</a>
  </div>
</div>
    <a href = "./assets/dashboard.html">
        <div class="col btn btn-danger" style="margin-left:70px; margin-
top:45px"> User Dashboard</div><div class="page-header">
    </a>
```

```
    <!--<div class="col btn btn-danger" style="margin-right:20px;right:0">
Story</div><div class="page-header"> -->
        </div>
        <div class="page-content jumbotron" style="  background-color:
red;
  background-image: linear-gradient(to right, darkgoldenrod , purple);">
  <h4 style="color:white; font-weight:bold;">Welcome to Account <?php
echo $username;?></h4></div>
  <span class="login-signup btn btn-warning"><a href="logout.php"
style="color:white">Logout</a></span>
        </div>
</BODY>
</HTML>
```

7.4 Database Schema :<u>MySQL</u>

- Database schema consist of the following field:
  ✓ id
  ✓ username
  ✓ password
  ✓ email
  ✓ create_at
- When the new user registers with their above details, it gets uploaded in the database.
- Now when the user login with username and password it checks whether the user has already registered or not. If it is a registered user it redirects to the home page or it shows the alert message.

Code:

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";
```

Retail Store Stock Inventory Analytics

```
/*!40101 SET
@OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET
@OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;


--
-- Database: `signup1`
--


-- --------------------------------------------------


--
-- Table structure for table `tbl_member`
--


CREATE TABLE `tbl_member` (
  `id` int(11) NOT NULL,
  `username` varchar(255) NOT NULL,
  `password` varchar(200) NOT NULL,
  `email` varchar(255) NOT NULL,
  `create_at` timestamp NOT NULL DEFAULT current_timestamp() ON
UPDATE current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=latin1;


--
-- Indexes for dumped tables
--
```

```
--
-- Indexes for table `tbl_member`
--
ALTER TABLE `tbl_member`
  ADD PRIMARY KEY (`id`);


--
-- AUTO_INCREMENT for dumped tables
--


--
-- AUTO_INCREMENT for table `tbl_member`
--
ALTER TABLE `tbl_member`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT
*/;
/*!40101 SET
CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET
COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

8. TESTING

8.1 Test Case:
- Verifies whether the user can login if he/she was an registered user.
- Verifies whether an unregistered user cannot proceed with the login.
- Verifies whether an unregistered user can successfully register as an user.

Retail Store Stock Inventory Analytics

- Verifies whether an register user cannot register them self as an new user.
- Verifies whether an alert message popsup when an unregistered user tries to login .
- Verifies whether an alert message popsup when an registered user tries to register again.
- Verifies whether an alert message popsup when an registered user enters his/her username or password incorrect.
- Verifies whether an alert message popsup when an new user registers.
- Verifies whether all UI button(signup,login now,logout,report,story,user dashboard) works efficiently.
- Verifies whether username popsup on the welcome note.

Test case ID	Feature Type	Compo nent	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Sta tus	Commnets	TC for Automation(Y/N)	BU G ID	Executed By
Testcase_1	Functional	Login Page	Verifies whether the user can login if he/she was an registered user	Checks whether the logged in username is registered in backend.	1.Enter your username 2.Enter your password 3.click signin button	username: Nalayia Thrian passwoed:abcd	Homepage should display	Working as expected	Pass				user
Testcase_2	Functional	Login Page	Verifies whether an unregistered user cannot proceed with the login.	Checks whether the logged in username is not registered in backend.	1.Enter your username 2.Enter your password 3.click signin button	username:nalayathrian password:1234	Homepage will not display	Working as expected	Pass				user
testcase_3	Functional	register page	Verifies whether an unregistered user can successfully register as an user.	The details given by the user is stored in backend	1.Enter your username 2.Enter your email 3.Enter your password 4.Enter your confirm password 5.Click on signup button	Enter your data	User will be able to access to login page	working as expected	Pass				user
testcase_4	Functional	Register page	Verifies whether an register user cannot register themself as an new user.	checks whether the user name is present in the database.	1.Enter your username 2.Enter your email 3.Enter your password 4.Enter your confirm password 5.Click on signup button	username:Nalayia Thrian password:abcd	User will not be able to access to login page	working as expected	Pass				user
Testcase_5	Functional	Login page	Verifies whether an alert message popsup when an unregistered user tries to login	checks whether the user name is present in the database.	1.Enter your username 2.Enter your password 3.click signin button	username:raja password:abcd	message should display	Working as expected	Pass				user

Testcase_6	Functional	Register page	Verifies whether an alert message popsup when an registered user tries to register again.	checks whether the user name is present in the database.	1.Enter your username 2.Enter your email 3.Enter your password 4.Enter your confirm password 5.Click on signup button	Username:Nalayia Thrian password: abcd	.message should display	Working as expected	Pass				user
Testcase_7	Functional	Login page	Verifies whether an alert message popsup when an registered user enters his/her username or password incorrect.	checks the username given by the user against the username and password in the database.	1.Enter your username 2.Enter your password 3.click signin button	Username:Nalayia Thrian password: 1234	user will not be able to access the home page	working as expected	Pass				user
Testcase_8	Functional	Register page	Verifies whether an alert message popsup when an new user registers.	checks whether the username and password is iploaded in the database.	1.Enter your username 2.Enter your email 3.Enter your password 4.Enter your confirm password 5.Click on signup button	Username:Nalayia Thrian password: 1234	message should display	Working as expected	Pass				user
Testcase_9	UI	Home page	Verifies whether all UI button(signup,loginow,logou t,report,story,user dashboard) works efficiently.	checks whether all the button are working efficiently	1.Enter username, password to login. 2.Click on report, story,user dashboard and logout button		button should work	Working as expected	Pass				user
Testcase_10	Functional	Home page	Verifies whether username popsup on the welcome note.	check whether the message display	1.Enter your username 2.Enter your password 3.click signin button		welcome note should display	Working as expected	Pass				user

8.2 User Acceptance Testing:

Retail Store Stock Inventory Analytics

The purpose of this is to briefly explain the test coverage and open issues of the retail store stock analytics project at the time of the release to User Acceptance Testing (UAT).

Defect Analysis:

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved.

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	8	4	2	1	15
Duplicate	0	0	0	0	0
External	3	2	0	1	6
Fixed	4	0	1	0	5
Not Reproduced	0	0	1	0	1
Skipped	0	0	0	1	1
Won't Fix	0	0	1	0	1
Totals	15	6	5	3	29

Test Case Analysis:

This report shows the number of test cases that have passed, failed,and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	5	0	0	5
Client Application	30	0	0	30
Security	2	0	0	2

Outsource Shipping	4	0	0	4
Exception Reporting	8	0	0	8
Final Report Output	6	0	0	6
Version Control	2	0	0	2

9. RESULTS

9.1 Performance Metrics:

Performance metrics are defined as figures and data representative of an organization's actions, abilities, and overall quality.

S.No.	Parameter	Screenshot / Values
1.	Dashboard design	Dashboard consist of 16 graph in 4 different tabs.
2.	Data Responsiveness	Data was responsive for creating dashboard, story and report.
3.	Amount Data to Rendered (DB2 Metrics)	Inventory management dataset which consist of 938 datas in it.
4.	Utilization of Data Filters	Data filters was used to find the top most of the data in form of visualization.
5.	Effective User Story	Story consist of 4 scenes and 5 graphs.
6.	Descriptive Reports	Created 2 reports with 7 graphs.

10. ADVANTAGES & DISADVANTAGES:

Advantage:

➤ An advantage of the retail inventory method is that it does not require a physical inventory.

➤ The retail inventory method only requires an organization to record the retail prices of inventory items.

Cost-Effective:

Manual inventory control would increase your labor and process costs.

Saves Time:

Paper-based retail inventory management can take a lot of time and effort.

Process Efficiency:

Inventory management is one of the crucial retail processes.

Disadvantage:

➤ Overstocking on products runs the risk of the product becoming obsolete.

➤ Higher storage and insurance costs.

➤ Certain goods might perish.

➤ Stock may become obsolete before it is used.

➤ Your capital is tied up

11. CONCLUSION:

Hence in Retail store stock analysis it helps shop holder to manage stock, sale and price and maintain the necessary stock without reaching to demand,by maintaining the stock it gains the trust for the customer to buy product on a regular basis which also provide gain to to shop holder by increasing the profit.

12. FUTURE SCOPE:

Inventory management systems have become more real-time, giving retailers more data about demographics, spending habits, shopping

preferences, etc.. Stock control for omni channel retailing.Stores doing omni channel retailing are at the top of their game; they attract the 90% of consumers who switch between at least three applications per day to complete specific tasks.Inventories that power experiential retail.

13. APPENDIX:

Source Code:

1)Home.php

```php
<?php
session_start();
if (isset($_SESSION["username"])) {
    $username = $_SESSION["username"];
    session_write_close();
} else {
    // since the username is not set in session, the user is not-logged-in
    // he is trying to access this page unauthorized
    // so let's clear all session variables and redirect him to index
    session_unset();
    session_write_close();
    $url = "./index.php";
    header("Location: $url");
}

?>
<HTML>
<HEAD>
<TITLE>Welcome</TITLE>
<link href="assets/css/phppot-style.css" type="text/css"
        rel="stylesheet" />
<link href="assets/css/user-registration.css" type="text/css"
```

```
        rel="stylesheet" />
    <!-- Latest compiled and minified CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" integrity="sha384-
BVYiiSlFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68v
bdEjh4u" crossorigin="anonymous">
<style>
    body{
        background-image:url("https://img.freepik.com/premium-
photo/magnifying-glass-documents-with-analytics-data-lying-table_33799-
7624.jpg?w=2000");
        background-position: center; /* Center the image */
  background-repeat: no-repeat; /* Do not repeat the image */
  background-size: cover; /* Resize the background image to cover the
entire container */
    }
.dropdown {
  position: absolute;
  display: inline-block;
}

.dropdown-content {
  display: none;
  position: absolute;
  background-color: #f1f1f1;
  min-width: 160px;
  z-index: 1;
}

.dropdown-content a {
```

```
    color: black;
    padding: 12px 16px;
    text-decoration: none;
    display: block;
}

.dropdown-content a:hover {background-color: #ddd}

.dropdown:hover .dropdown-content {
    display: block;
}

.btn:hover, .dropdown:hover .btn {
    background-color: #6b2139d6;
}
</style>
</HEAD>
<BODY>
    <!-- <img src="./assets/bg.jpg" alt=""> -->
        <div class="phppot-container">
    <a href = "./assets/story.html">
    <div class="col btn btn-danger" style="margin-top:45px;right: 0">
Story</div>
    </a>


    <div class="dropdown">
      <div class="col btn btn-danger" style="margin-top:45px;right: 0">
Report</div>
      </button>
    <div class="dropdown-content">
```

```html
    <a href="./assets/report1.html">Report 1</a>
    <a href="./assets/report2.html">Report 2</a>
  </div>
</div>
  <a href = "./assets/dashboard.html">
    <div class="col btn btn-danger" style="margin-left:70px; margin-top:45px"> User Dashboard</div><div class="page-header">
  </a>

    <!--<div class="col btn btn-danger" style="margin-right:20px;right:0">
Story</div><div class="page-header"> -->
            </div>
            <div class="page-content jumbotron" style="  background-color: red;
  background-image: linear-gradient(to right, darkgoldenrod , purple);">
  <h4 style="color:white; font-weight:bold;">Welcome to Account <?php
echo $username;?></h4></div>
  <span class="login-signup btn btn-warning"><a href="logout.php"
style="color:white">Logout</a></span>
        </div>
</BODY>
</HTML>
```

2)Index.php
```php
<?php
require_once __DIR__ . "/login.php";
```
3)Login.php
```php
<?php
use Phppot\Member;

if (! empty($_POST["login-btn"])) {
```

```php
    require_once __DIR__ . '/Model/Member.php';
    $member = new Member();
    $loginResult = $member->loginMember();
}
?>
```

```html
<HTML>
<HEAD>
<TITLE>Login</TITLE>
<link href="assets/css/phppot-style.css" type="text/css"
      rel="stylesheet" />
<link href="assets/css/user-registration.css" type="text/css"
      rel="stylesheet" />
<script src="vendor/jquery/jquery-3.3.1.js" type="text/javascript"></script>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.mi
n.css" integrity="sha384-
BVYiiSlFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68v
bdEjh4u" crossorigin="anonymous">
<style>
        body{
    background-
image:url("https://media.istockphoto.com/id/913219882/photo/financial-
graph-on-technology-abstract-
background.jpg?b=1&s=170667a&w=0&k=20&c=aFXB_2Bn_T7weuTYIqZAs
8sIja9MHcT-u-TlBVftvqo=");
    background-position: center; /* Center the image */
  background-repeat: no-repeat; /* Do not repeat the image */
  background-size: cover; /* Resize the background image to cover the
entire container */
    }
.sign-up-container{
```

Retail Store Stock Inventory Analytics

```
  background-image: linear-gradient(to right, #ff0030 , #790975);
}
.form-label{
color:white !important;
}
#login-btn{
        color:white;
        font-weight:bold;
        background: #343a40;
}
</style>
</HEAD>
<BODY>
        <div class="phppot-container">
                <div class="sign-up-container">
                        <div class="login-signup">
                                <a href="user-registration.php"
style="color:white;">Sign up</a>
                        </div>
                        <div class="signup-align">
                                <form name="login" action="" method="post"
                                        onsubmit="return loginValidation()">
                                        <div class="signup-heading"
style="color:white;">Login</div>
                                        <?php if(!empty($loginResult)){?>
                                        <div class="error-msg"><?php echo
$loginResult;?></div>
                                        <?php }?>
                                        <div class="row">
                                                <div class="inline-block">
                                                        <div class="form-label">
```

Retail Store Stock Inventory Analytics

Username
					</div>
					<input class="input-box-330" type="text" name="username"
						id="username">
				</div>
			</div>
			<div class="row">
				<div class="inline-block">
					<div class="form-label">
						Password
					</div>
					<input class="input-box-330" type="password"
						name="login-password" id="login-password">
				</div>
			</div>
			<div class="row">
				<input class="btn btn-dark" type="submit" name="login-btn"
					id="login-btn" value="Login Now">
			</div>
		</form>
	</div>
	</div>
</div>

<script>

Retail Store Stock Inventory Analytics

```
function loginValidation() {
        var valid = true;
        $("#username").removeClass("error-field");
        $("#password").removeClass("error-field");

        var UserName = $("#username").val();
        var Password = $('#login-password').val();

        $("#username-info").html("").hide();

        if (UserName.trim() == "") {
                $("#username-info").html("required.").css("color",
"#ee0000").show();
                $("#username").addClass("error-field");
                valid = false;
        }
        if (Password.trim() == "") {
                $("#login-password-info").html("required.").css("color",
"#ee0000").show();
                $("#login-password").addClass("error-field");
                valid = false;
        }
        if (valid == false) {
                $('.error-field').first().focus();
                valid = false;
        }
        return valid;
}
</script>
</BODY>
</HTML>
```

4)logout.php

```php
<?php
// clear all the session variables and redirect to index
session_start();
session_unset();
session_write_close();
$url = "./index.php";
header("Location: $url");
```

5)User-registration.php

```php
<?php
use Phppot\Member;
if (! empty($_POST["signup-btn"])) {
    require_once './Model/Member.php';
    $member = new Member();
    $registrationResponse = $member->registerMember();
}
?>
<HTML>
<HEAD>
<TITLE>User Registration</TITLE>
<link href="assets/css/phppot-style.css" type="text/css"
    rel="stylesheet" />
<link href="assets/css/user-registration.css" type="text/css"
    rel="stylesheet" />
<script src="vendor/jquery/jquery-3.3.1.js" type="text/javascript"></script>
</HEAD>
<style>
        body{
    background-
image:url("https://i.pinimg.com/736x/1a/f1/c6/1af1c62ecf24b3d90a9f4f2
```

```
05914e27e--business-analyst-data-analytics.jpg");
      background-position: center; /* Center the image */
  background-repeat: no-repeat; /* Do not repeat the image */
  background-size: cover; /* Resize the background image to cover the
entire container */
    }
.sign-up-container{
  background-image: linear-gradient(to right, #790975,#ff0030);
}
.form-label{
color:white !important;
}
#signup-btn{
      color:white;
      font-weight:bold;
      background: #343a40;
}
</style>
<BODY>

        <div class="phppot-container">
              <div class="sign-up-container">
                    <div class="login-signup">
                          <a href="index.php" style="color:white">Login</a>
                    </div>
                    <div class="">
                          <form name="sign-up" action="" method="post"
                                onsubmit="return signupValidation()">
                                <div class="signup-heading"
style="color:white">Registration</div>
                                <?php
```

Retail Store Stock Inventory Analytics

```php
if (! empty($registrationResponse["status"])) {
    ?>
            <?php
    if ($registrationResponse["status"] == "error") {
        ?>
                            <div class="server-response error-msg"><?php
echo $registrationResponse["message"]; ?></div>
            <?php
    } else if ($registrationResponse["status"] == "success") {
        ?>
            <div class="server-response success-msg"><?php echo
$registrationResponse["message"]; ?></div>
            <?php
    }
    ?>
                    <?php
}
?>  <div class="error-msg" id="error-msg"></div>
                        <div class="row">
                            <div class="inline-block">
                                <div class="form-label">
                                    Username<span
class="required error" id="username-info"></span>
                                </div>
                                <input class="input-box-330"
type="text" name="username"
                                    id="username">
                            </div>
                        </div>
                        <div class="row">
                            <div class="inline-block">
```

```
                                              <div class="form-label">
                                                      Email<span class="required
error" id="email-info"></span>
                                              </div>
                                              <input class="input-box-330"
type="email" name="email" id="email">
                                      </div>
                              </div>
                              <div class="row">
                                      <div class="inline-block">
                                              <div class="form-label">
                                                      Password<span
class="required error" id="signup-password-info"></span>
                                              </div>
                                              <input class="input-box-330"
type="password"

                                                      name="signup-password"
id="signup-password">
                                      </div>
                              </div>
                              <div class="row">
                                      <div class="inline-block">
                                              <div class="form-label">
                                                      Confirm Password<span
class="required error"

                                                              id="confirm-password-
info"></span>
                                              </div>
                                              <input class="input-box-330"
type="password"

                                                      name="confirm-password"
```

```
id="confirm-password">
                                        </div>
                                </div>
                                <div class="row">
                                        <input class="btn" type="submit"
name="signup-btn"
                                                        id="signup-btn" value="Sign up">
                                </div>
                        </form>
                </div>
        </div>
    </div>
    <script>
function signupValidation() {
        var valid = true;
        $("#username").removeClass("error-field");
        $("#email").removeClass("error-field");
        $("#password").removeClass("error-field");
        $("#confirm-password").removeClass("error-field");
        var UserName = $("#username").val();
        var email = $("#email").val();
        var Password = $('#signup-password').val();
    var ConfirmPassword = $('#confirm-password').val();
        var emailRegex = /^[a-zA-Z0-9.!#$%&'*+/=?^_`{|}~-]+@[a-zA-Z0-9](?:[a-
zA-Z0-9-]{0,61}[a-zA-Z0-9])?(?:\.[a-zA-Z0-9](?:[a-zA-Z0-9-]{0,61}[a-zA-Z0-
9])?)*$/;
        $("#username-info").html("").hide();
        $("#email-info").html("").hide();
        if (UserName.trim() == "") {
                $("#username-info").html("required.").css("color",
"#ee0000").show();
```

```
        $("#username").addClass("error-field");
        valid = false;
}
if (email == "") {
        $("#email-info").html("required").css("color", "#ee0000").show();
        $("#email").addClass("error-field");
        valid = false;
} else if (email.trim() == "") {
        $("#email-info").html("Invalid email address.").css("color",
"#ee0000").show();
        $("#email").addClass("error-field");
        valid = false;
} else if (!emailRegex.test(email)) {
        $("#email-info").html("Invalid email address.").css("color",
"#ee0000")
                        .show();
        $("#email").addClass("error-field");
        valid = false;
}
if (Password.trim() == "") {
        $("#signup-password-info").html("required.").css("color",
"#ee0000").show();
        $("#signup-password").addClass("error-field");
        valid = false;
}
if (ConfirmPassword.trim() == "") {
        $("#confirm-password-info").html("required.").css("color",
"#ee0000").show();
        $("#confirm-password").addClass("error-field");
        valid = false;
}
```

```
        if(Password != ConfirmPassword){
      $("#error-msg").html("Both passwords must be same.").show();
        valid=false;
    }
        if (valid == false) {
                $('.error-field').first().focus();
                valid = false;
        }
        return valid;
}
</script>
</BODY>
</HTML>
```

6)DataSource.php

```php
<?php

namespace Phppot;

class DataSource
{


    const HOST = 'localhost';

    const USERNAME = 'root';

    const PASSWORD = '';

    const DATABASENAME = 'signup1';

    private $conn;
```

```php
function __construct()
{
    $this->conn = $this->getConnection();
}


public function getConnection()
{
    $conn = new \mysqli(self::HOST, self::USERNAME, self::PASSWORD,
self::DATABASENAME);

    if (mysqli_connect_errno()) {
        trigger_error("Problem with connecting to database.");
    }

    $conn->set_charset("utf8");
    return $conn;
}


public function getPdoConnection()
{
    $conn = FALSE;
    try {
        $dsn = 'mysql:host=' . self::HOST . ';dbname=' .
self::DATABASENAME;
        $conn = new \PDO($dsn, self::USERNAME, self::PASSWORD);
        $conn->setAttribute(\PDO::ATTR_ERRMODE,
\PDO::ERRMODE_EXCEPTION);
```

Retail Store Stock Inventory Analytics

```php
    } catch (\Exception $e) {
        exit("PDO Connect Error: " . $e->getMessage());
    }
    return $conn;
}


public function select($query, $paramType = "", $paramArray = array())
{
    $stmt = $this->conn->prepare($query);

    if (! empty($paramType) && ! empty($paramArray)) {

        $this->bindQueryParams($stmt, $paramType, $paramArray);
    }
    $stmt->execute();
    $result = $stmt->get_result();

    if ($result->num_rows > 0) {
        while ($row = $result->fetch_assoc()) {
            $resultset[] = $row;
        }
    }

    if (! empty($resultset)) {
        return $resultset;
    }
}


public function insert($query, $paramType, $paramArray)
```

Retail Store Stock Inventory Analytics

```php
{
    $stmt = $this->conn->prepare($query);
    $this->bindQueryParams($stmt, $paramType, $paramArray);

    $stmt->execute();
    $insertId = $stmt->insert_id;
    return $insertId;
}


public function execute($query, $paramType = "", $paramArray = array())
{
    $stmt = $this->conn->prepare($query);

    if (! empty($paramType) && ! empty($paramArray)) {
        $this->bindQueryParams($stmt, $paramType, $paramArray);
    }
    $stmt->execute();
}


    public function bindQueryParams($stmt, $paramType, $paramArray = array())
    {
        $paramValueReference[] = & $paramType;
        for ($i = 0; $i < count($paramArray); $i ++) {
            $paramValueReference[] = & $paramArray[$i];
        }
        call_user_func_array(array(
            $stmt,
            'bind_param'
```

```php
    ), $paramValueReference);
  }


  public function getRecordCount($query, $paramType = "", $paramArray =
array())
  {
    $stmt = $this->conn->prepare($query);
    if (! empty($paramType) && ! empty($paramArray)) {

      $this->bindQueryParams($stmt, $paramType, $paramArray);
    }
    $stmt->execute();
    $stmt->store_result();
    $recordCount = $stmt->num_rows;

    return $recordCount;
  }
}
```

7)Member.php

```php
<?php
namespace Phppot;

class Member
{

  private $ds;

  function __construct()
  {
    require_once __DIR__ . '/../lib/DataSource.php';
```

```php
        $this->ds = new DataSource();
    }


public function isUsernameExists($username)
{
    $query = 'SELECT * FROM tbl_member where username = ?';
    $paramType = 's';
    $paramValue = array(
        $username
    );
    $resultArray = $this->ds->select($query, $paramType, $paramValue);
    $count = 0;
    if (is_array($resultArray)) {
        $count = count($resultArray);
    }
    if ($count > 0) {
        $result = true;
    } else {
        $result = false;
    }
    return $result;
}


public function isEmailExists($email)
{
    $query = 'SELECT * FROM tbl_member where email = ?';
    $paramType = 's';
    $paramValue = array(
        $email
```

```php
    );
    $resultArray = $this->ds->select($query, $paramType, $paramValue);
    $count = 0;
    if (is_array($resultArray)) {
        $count = count($resultArray);
    }
    if ($count > 0) {
        $result = true;
    } else {
        $result = false;
    }
    return $result;
}


public function registerMember()
{
    $isUsernameExists = $this->isUsernameExists($_POST["username"]);
    $isEmailExists = $this->isEmailExists($_POST["email"]);
    if ($isUsernameExists) {
        $response = array(
            "status" => "error",
            "message" => "Username already exists."
        );
    } else if ($isEmailExists) {
        $response = array(
            "status" => "error",
            "message" => "Email already exists."
        );
    } else {
        if (! empty($_POST["signup-password"])) {
```

```php
        // PHP's password_hash is the best choice to use to store
passwords
        // do not attempt to do your own encryption, it is not safe
        $hashedPassword = password_hash($_POST["signup-password"],
PASSWORD_DEFAULT);
    }
    $query = 'INSERT INTO tbl_member (username, password, email)
VALUES (?, ?, ?)';
    $paramType = 'sss';
    $paramValue = array(
        $_POST["username"],
        $hashedPassword,
        $_POST["email"]
    );
    $memberId = $this->ds->insert($query, $paramType, $paramValue);
    if (! empty($memberId)) {
        $response = array(
            "status" => "success",
            "message" => "You have registered successfully."
        );
    }
  }
  return $response;
}

public function getMember($username)
{
  $query = 'SELECT * FROM tbl_member where username = ?';
  $paramType = 's';
  $paramValue = array(
```

```php
        $username
    );
    $memberRecord = $this->ds->select($query, $paramType,
$paramValue);
    return $memberRecord;
  }



  public function loginMember()
  {
    $memberRecord = $this->getMember($_POST["username"]);
    $loginPassword = 0;
    if (! empty($memberRecord)) {
      if (! empty($_POST["login-password"])) {
        $password = $_POST["login-password"];
      }
      $hashedPassword = $memberRecord[0]["password"];
      $loginPassword = 0;
      if (password_verify($password, $hashedPassword)) {
        $loginPassword = 1;
      }
    } else {
      $loginPassword = 0;
    }
    if ($loginPassword == 1) {
      // login sucess so store the member's username in
      // the session
      session_start();
      $_SESSION["username"] = $memberRecord[0]["username"];
      session_write_close();
      $url = "./home.php";
```

```
        header("Location: $url");
    } else if ($loginPassword == 0) {
        $loginStatus = "Invalid username or password.";
        return $loginStatus;
    }
  }
}
```

8)user-registration.sql

```
-- phpMyAdmin SQL Dump
-- version 5.1.1
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: Oct 13, 2021 at 08:09 PM
-- Server version: 10.4.20-MariaDB
-- PHP Version: 7.3.29


SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";



/*!40101 SET
@OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET
@OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
```

Retail Store Stock Inventory Analytics

```
--
-- Database: `signup1`
--


-- --------------------------------------------------


--
-- Table structure for table `tbl_member`
--

CREATE TABLE `tbl_member` (
  `id` int(11) NOT NULL,
  `username` varchar(255) NOT NULL,
  `password` varchar(200) NOT NULL,
  `email` varchar(255) NOT NULL,
  `create_at` timestamp NOT NULL DEFAULT current_timestamp() ON
UPDATE current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Indexes for dumped tables
--


--
-- Indexes for table `tbl_member`
--
ALTER TABLE `tbl_member`
  ADD PRIMARY KEY (`id`);

--
-- AUTO_INCREMENT for dumped tables
```

Retail Store Stock Inventory Analytics

```
--

--
-- AUTO_INCREMENT for table `tbl_member`
--
ALTER TABLE `tbl_member`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT
*/;
/*!40101 SET
CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET
COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

9)phppot-style.css
```
/* version 3.0 table added */
.phppot-container {
      -webkit-font-smoothing: antialiased;
      font-family: Arial, "Helvetica Neue", Helvetica, sans-serif;
      font-size: .9em;
      color: #1e2a28;
      width: 740px;
      margin: 0 auto;
      padding: 0px 20px 20px 20px;
}

.phppot-container table {
```

```css
        border-collapse: collapse;
        width: 100%;
        margin-bottom: 30px;
}

.phppot-container td, .phppot-container th {
        text-align: left;
        padding: 8px;
}

.phppot-container h1 {
        font-weight: normal;
}

input[type=text].phppot-input, .phppot-container input,
        .phppot-container textarea, .phppot-container select,
.phppot-input,
        .phppot-select {
        box-sizing: border-box;
        width: 200px;
        height: initial;
        padding: 8px 5px;
        border: 1px solid #9a9a9a;
        border-radius: 3px;
}

.phppot-container input[type="checkbox"] {
```

```css
        width: auto;
        vertical-align: text-bottom;
        display: initial;
        opacity: initial;
        position: inherit;
        pointer-events: initial;
}

.phppot-container textarea, .phppot-textarea {
        width: 300px;
}

.phppot-container select, .phppot-select {
        display: initial;
        height: 30px;
        background-color: #fff;
        padding: 2px 5px;
}

.phppot-container button, .phppot-container input[type=submit] {
        padding: 8px 0px;
        font-size: 1em;
        cursor: pointer;
        border-radius: 3px;
        color: #565656;
        font-weight: bold;
        background-color: #ffc72c;
```

```
        border-color: #ffd98e #ffbe3d #de9300;
}

.phppot-container button, .phppot-container
input[type=submit]:hover {
        background-color: #f7c027;
}

.phppot-container button:focus {
        outline: none;
}

.phppot-container .phppot-row {
        padding-top: 15px;
}

#phppot-message {
        padding: 6px 20px;
        font-size: 1em;
        color: rgb(40, 40, 40);
        box-sizing: border-box;
        margin: 0px;
        border-radius: 3px;
        width: 100%;
        overflow: auto;
}
```

Retail Store Stock Inventory Analytics

```css
.phppot-container .error {
      padding: 6px 20px;
      border-radius: 3px;
      background-color: #fb817c;
      border: 1px solid #e46b66;
}

.phppot-container .success {
      background-color: #48e0a4;
      border: #40cc94 1px solid;
}

#phppot-loader-icon {
      color: #1871e6;
      font-weight: bold;
      padding: 6px 20px 6px 0;
      display: none;
      vertical-align: middle;
}

#phppot-loader-ack-icon {
      color: #1871e6;
      font-weight: bold;
      padding: 6px 20px 6px 0;
      display: none;
      vertical-align: middle;
}
```

```
#phppot-btn-send:hover {
      background: #1363cc;
      border: #105bbd 1px solid;
}


.phppot-container .validation-message {
      color: #e20900;
      display: inline-block;
}


.phppot-container .label {
      margin-bottom: 3px;
}


.phppot-form, .phppot-section {
      border: #eaeaea 1px solid;
      padding: 10px 25px 10px 30px;
      border-radius: 3px;
}


.phppot-container .display-none {
      display: none;
}


.icon-add-more-attachemnt {
      cursor: pointer;
```

```
        font-style: italic;
        font-size: .9em;
}

.inline-block {
        display: inline-block;
}

@media all and (max-width: 780px) {
        .phppot-container {
                width: auto;
        }
}

@media all and (max-width: 400px) {
        .phppot-container {
                padding: 0px 20px;
        }
        .phppot-container h1 {
                font-size: 1.2em;
        }
        .phppot-container input, .phppot-container textarea, .phppot-
container select
                {
                width: 100%;
        }
        .phppot-form {
```

```
            border: none;
            padding: 0;
        }
}

dropdown {
        position: absolute;
        display: inline-block;
 }

 .dropdown-content {
        display: none;
        position: absolute;
        background-color: #f1f1f1;
        min-width: 160px;
        z-index: 1;
 }

 .dropdown-content a {
        color: black;
        padding: 12px 16px;
        text-decoration: none;
        display: block;
 }

 .dropdown-content a:hover {background-color: #ddd}
```

```css
.dropdown:hover .dropdown-content {
    display: block;
}


.btn:hover, .dropdown:hover .btn {
    background-color: #6b2139d6;
}
```

10)user-registration.css

```css
.sign-up-container {
    border: 1px solid;
    border-color: #9a9a9a;
    background: #fff;
    border-radius: 4px;
    padding: 10px;
    width: 350px;
    margin: 50px auto;
}

.page-header {
    float: right;
}

.login-signup {
    margin: 10px;
    text-decoration: none;
    float: right;
}
```

Retail Store Stock Inventory Analytics

```css
.login-signup a {
        text-decoration: none;
        font-weight: 700;
}

.signup-heading {
        font-size: 2em;
        font-weight: bold;
        padding-top: 60px;
        text-align: center;
}

.inline-block {
        display: inline-block;
}

.row {
        margin: 15px 0px;
        text-align: center;
}

.form-label {
        margin-bottom: 5px;
        text-align: left;
}
```

Retail Store Stock Inventory Analytics

```css
input.input-box-330 {
    width: 250px;
}

.sign-up-container .error {
    color: #ee0000;
    padding: 0px;
    background: none;
    border: #ee0000;
}

.sign-up-container .error-field {
    border: 1px solid #d96557;
}

.sign-up-container .error:before {
    content: '*';
    padding: 0 3px;
    color: #D8000C;
}

.error-msg {
    padding-top: 10px;
    color: #D8000C;
    text-align: center;
}
```

```css
.success-msg {
    padding-top: 10px;
        color: #176701;
        text-align: center;
}

input.btn {
        width: 250px
}

.signup-align {
        margin: 0 auto;
}

.page-content {
        font-weight: bold;
        padding-top: 60px;
        text-align: center;
}
```

11)dashboard.html
```html
<html>
<head>
<body>
<p><<iframe
src="https://us3.ca.analytics.ibm.com/bi/?perspective=dashboard&amp;pathRef=.my_folders%2FInventory%2BDashboard&amp;closeWindowOnLastView=true&amp;ui_appbar=false&amp;ui_navb
```

ar=false&shareMode=embedded&action=view&mode=dashboard&subView=model0000018442eaff4e_00000000
0" width="1300" height="750" frameborder="0" gesture="media" allow="encrypted-media" allowfullscreen=""></iframe></p>
</head>
</html>

12)story.html

<html>
<head>
<body>
<p><iframe
src="https://us3.ca.analytics.ibm.com/bi/?perspective=story&pathRef=.my_folders%2FRetail%2BStory&closeWindowOnLastView=true&ui_appbar=false&ui_navbar=false&shareMode=embedded&action=view&sceneId=model000001846b72d9c8_00000000&sceneTime=0" width="1300" height="750" frameborder="0" gesture="media" allow="encrypted-media" allowfullscreen=""></iframe></p>
</head>
</html>

13)report1.html

<html>
<head>
<body>
<p><iframe
src="https://us3.ca.analytics.ibm.com/bi/?pathRef=.my_folders%2FRetail%2BStock%2BReport&closeWindowOnLastView=tru

e&ui_appbar=false&ui_navbar=false&shareMode=e
mbedded&action=run&format=HTML&prompt=fals
e" width="1300" height="750" frameborder="0" gesture="media"
allow="encrypted-media" allowfullscreen=""></iframe></p>
</head>
</html>
14)report2.html
<html>
<head>
<body>
<p><iframe
src="https://us3.ca.analytics.ibm.com/bi/?pathRef=.my_folders%
2FNew%2Breport2&closeWindowOnLastView=true&ui_a
ppbar=false&ui_navbar=false&shareMode=embedded&
amp;action=run&format=HTML&prompt=false"
width="1300" height="700" frameborder="0" gesture="media"
allow="encrypted-media" allowfullscreen=""></iframe>
</p>
</head>
</html>
GitHub & Project Demo Link:
GitHub Link: https://github.com/IBM-EPBL/IBM-Project-21730-
1659789475
Project Demo Link:
https://drive.google.com/file/d/1kDvREPuxp77vcFuJL9L7pzAXTA
FhQmms/view?usp=sharing