Retail Store Stock Inventory Analytics

1. INTRODUCTION

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.

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

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.

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.

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 favorite electronic gadgets.	Most of the products are not available in store	But the demand for the product is high	Displeased
PS-2	A Foodie	Buy my favorite dish	It is not available in most of the hotels	It is a Continent-al dish	Frustrated

Retail Store Stock Inventory Analytics

Problem Statement-1:

A Shopaholic →Buy my favorite electronic gadgets→Not available→High Demand→Displeased

Problem Statement-2:

A Foodie→Buy my favorite dish→Not available→Continental dish→Frustated

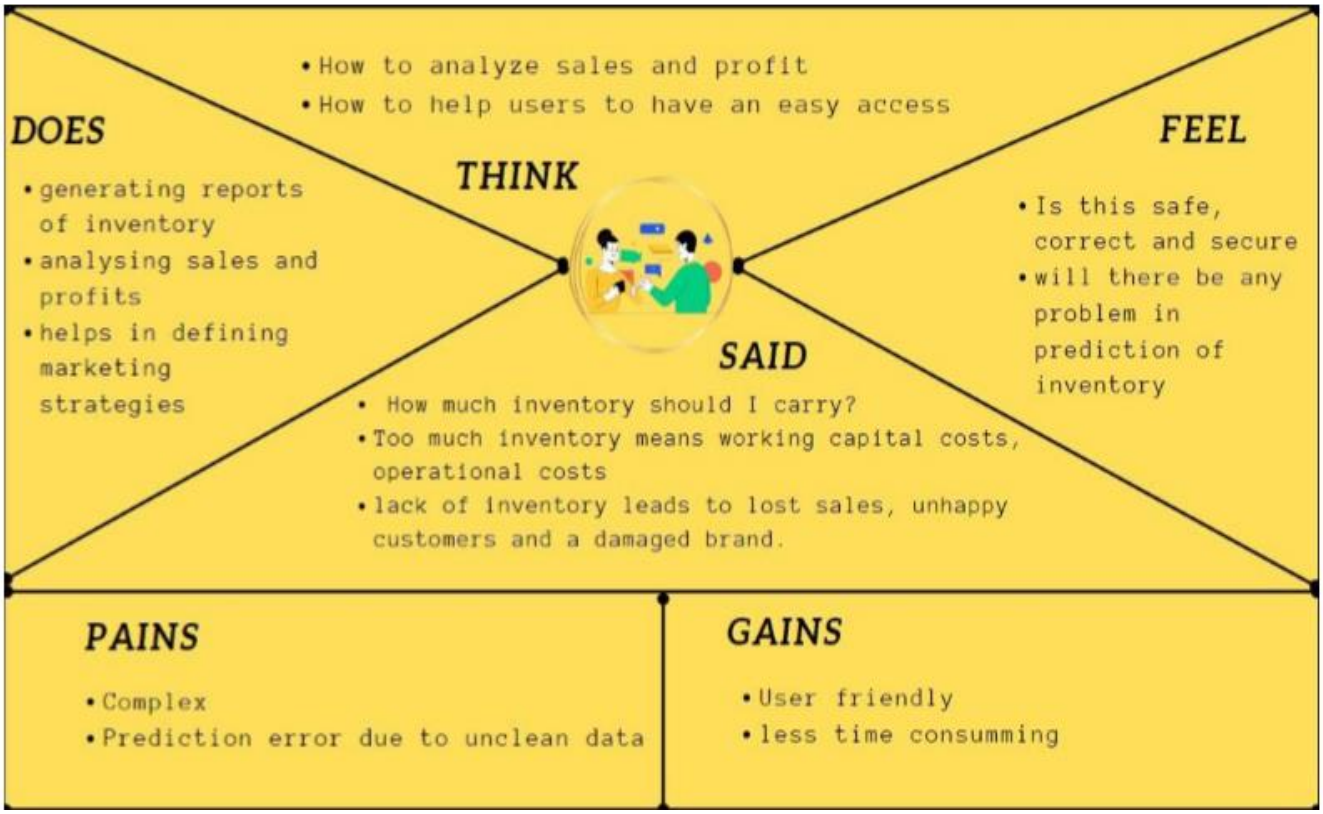
3. IDEATION & PROPOSED SOLUTION

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 reflectfour 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

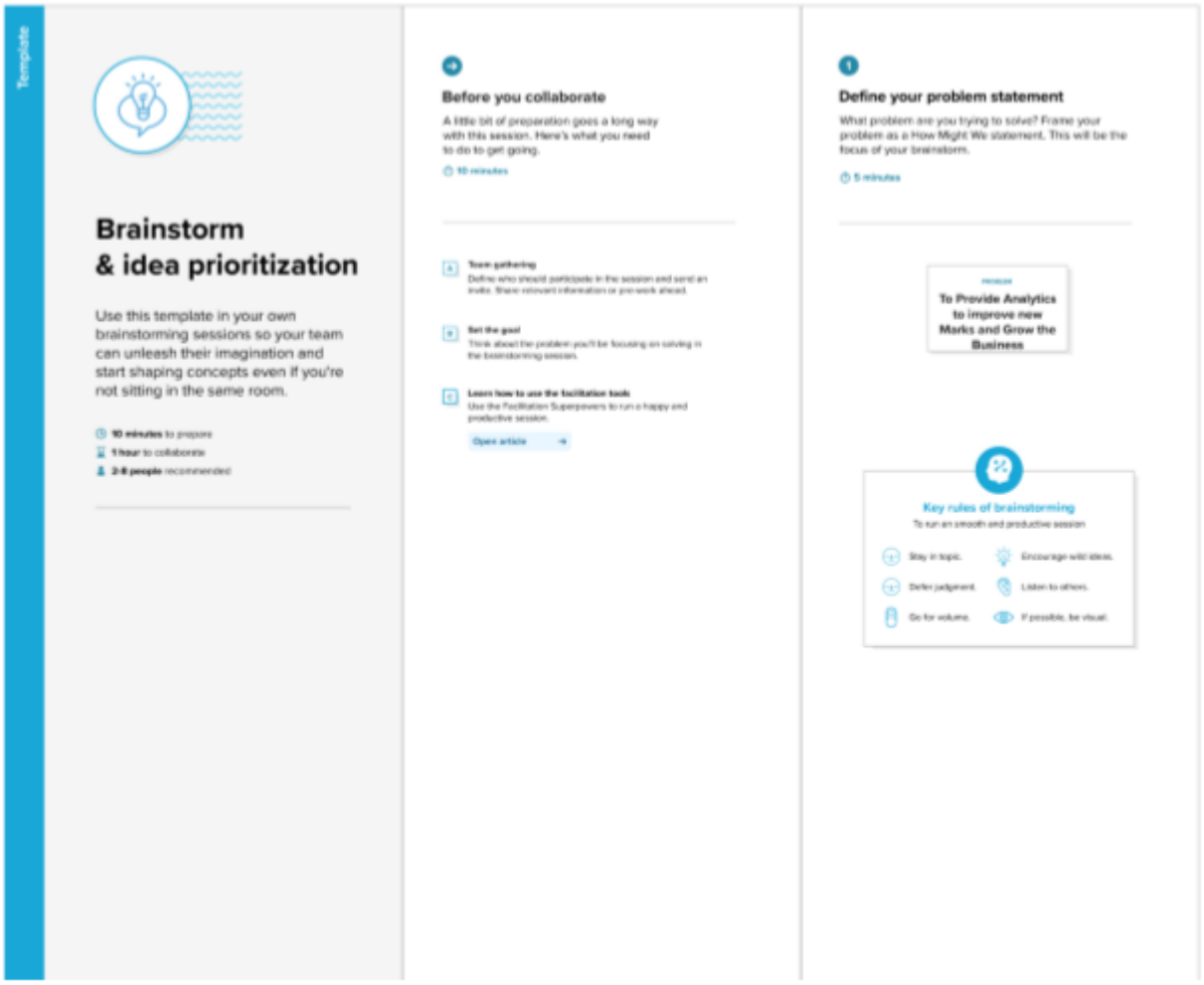


Ideation & Brainstorming:

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.

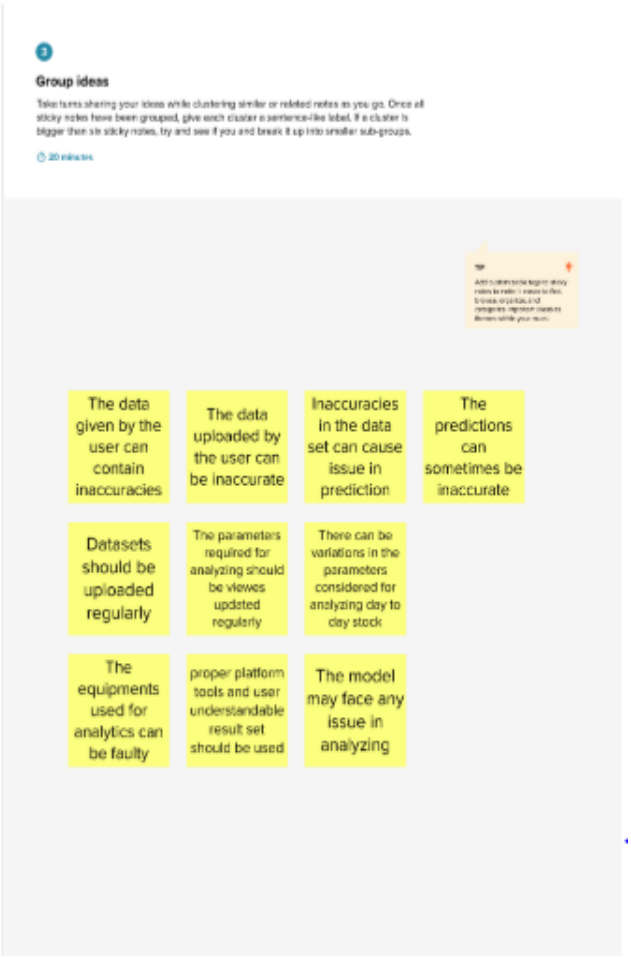
Retail Store Stock Inventory Analytics

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



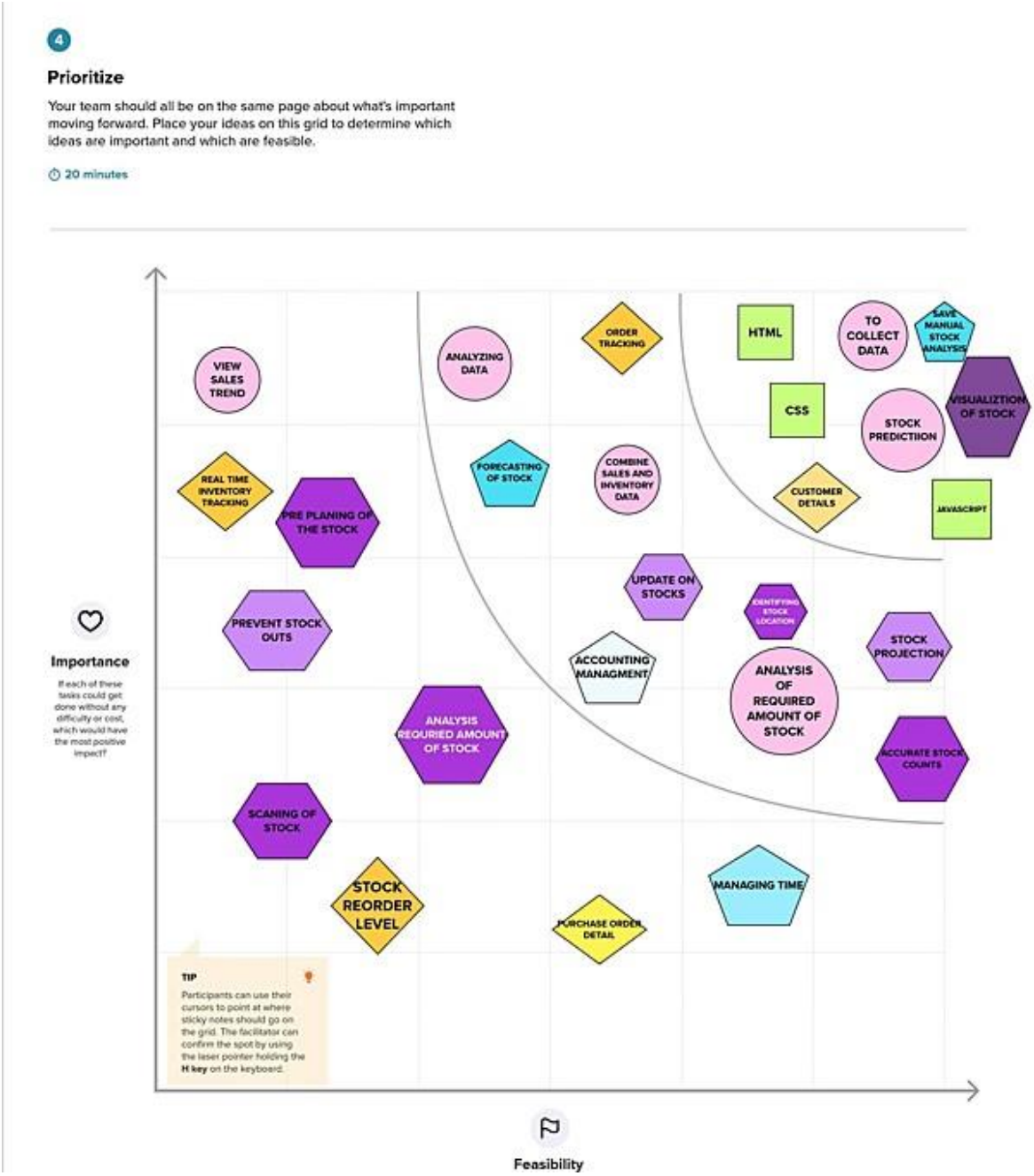
Retail Store Stock Inventory Analytics

Step-2: Brainstorm, Idea Listing and Grouping



Retail Store Stock Inventory Analytics

Step-3: Idea Prioritization



Retail Store Stock Inventory Analytics

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.

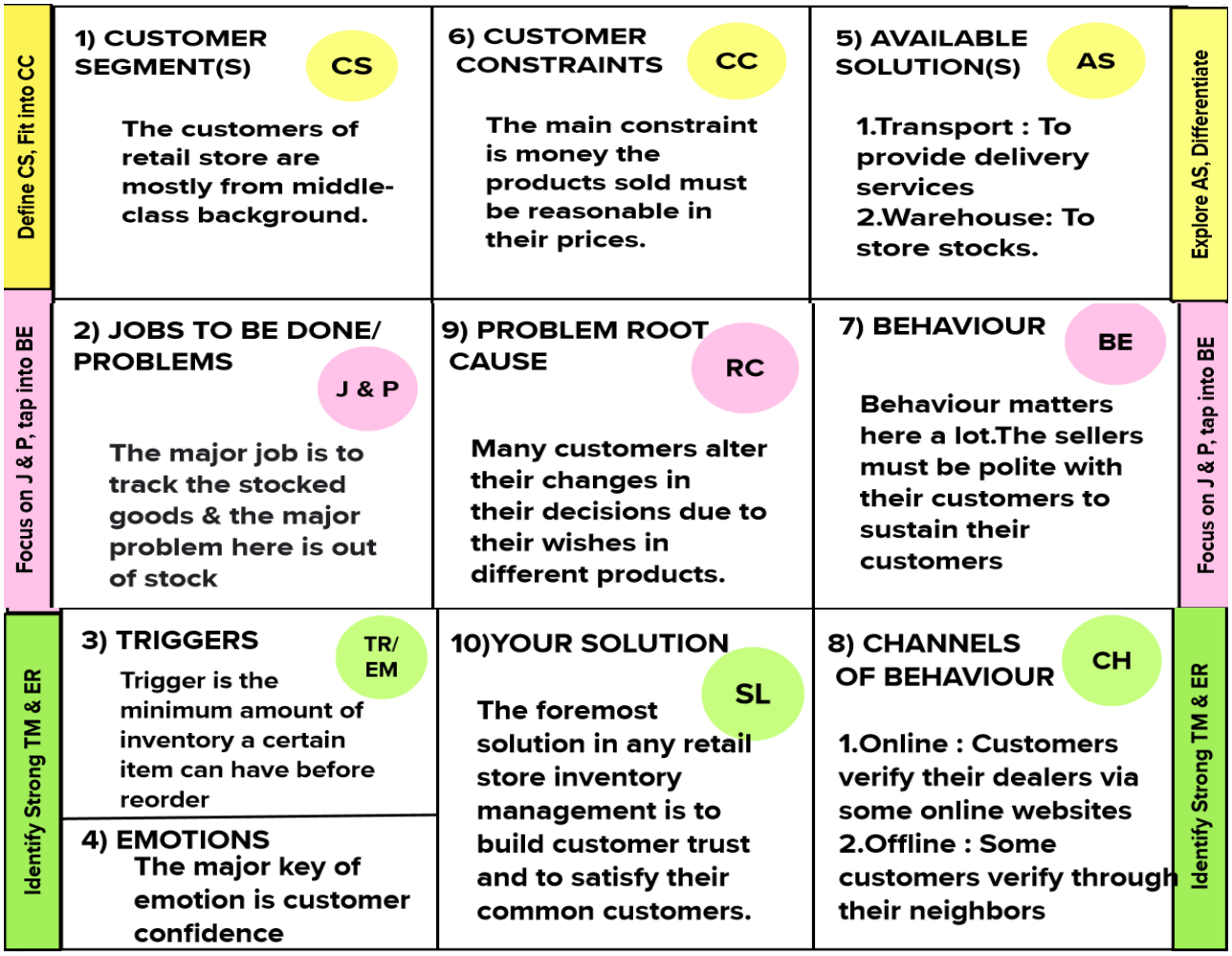
6.	Scalability of the Solution	When your inventory is hard to identify or locate in the warehouse, it leads to incomplete, inaccurate or delayed shipments. Receiving and finding the right stock is vital to efficient warehouse operations and provides a positive customer experiences.

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 andget a better overview of current situation.

Retail Store Stock Inventory Analytics



4. REQUIREMENT ANALYSIS

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.

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

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

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.

Retail Store Stock Inventory Analytics

Solution & Technical Architecture:

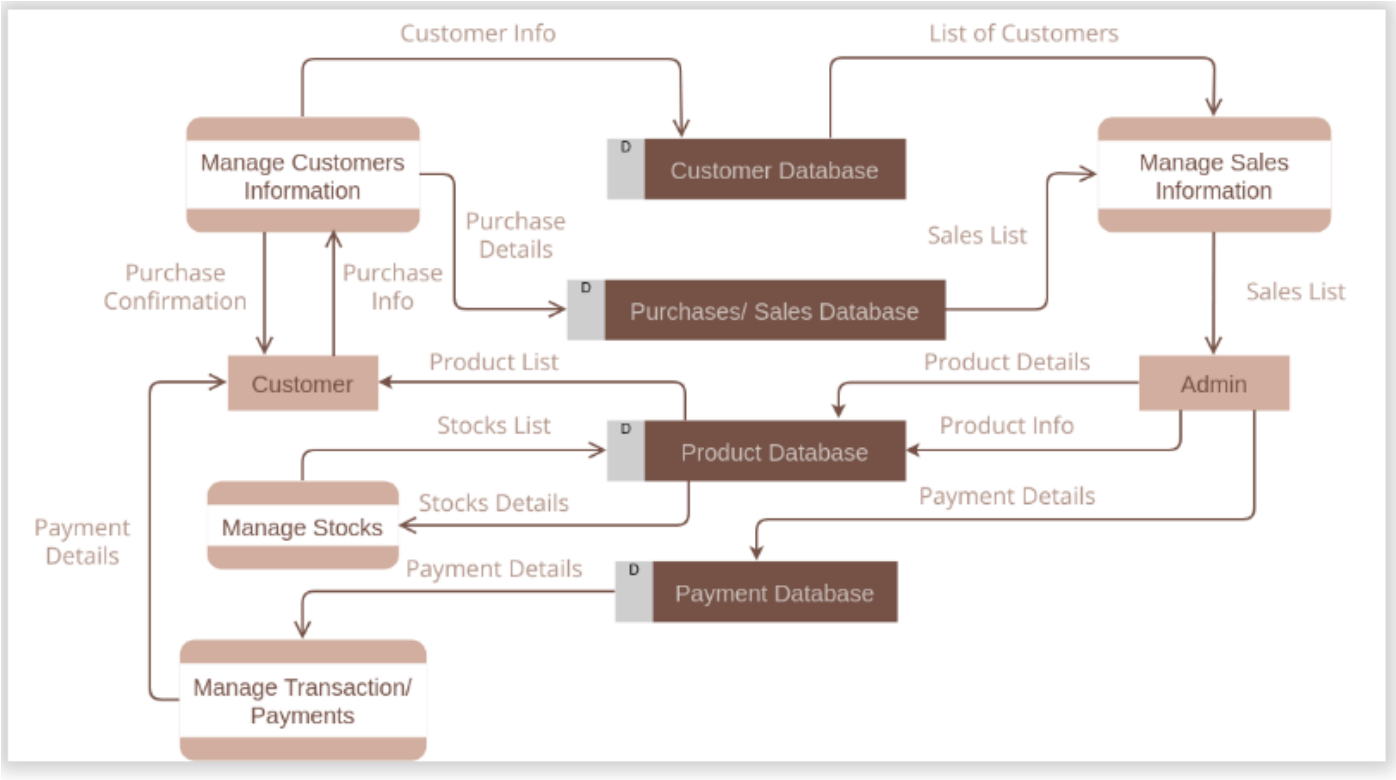
Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.

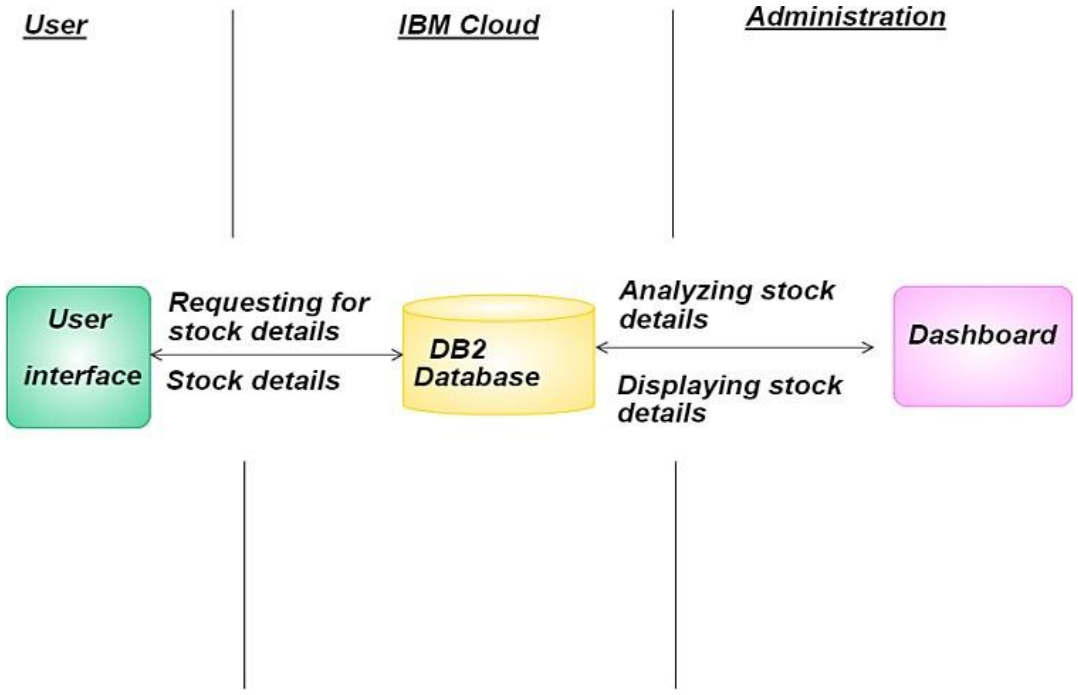
Retail Store Stock Inventory Analytics

- 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



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.



User Stories:

A user story is an informal, general explanation of a software featurewritten 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 valueback 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

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

Retail Store Stock Inventory Analytics

Sprint Delivery Schedule:

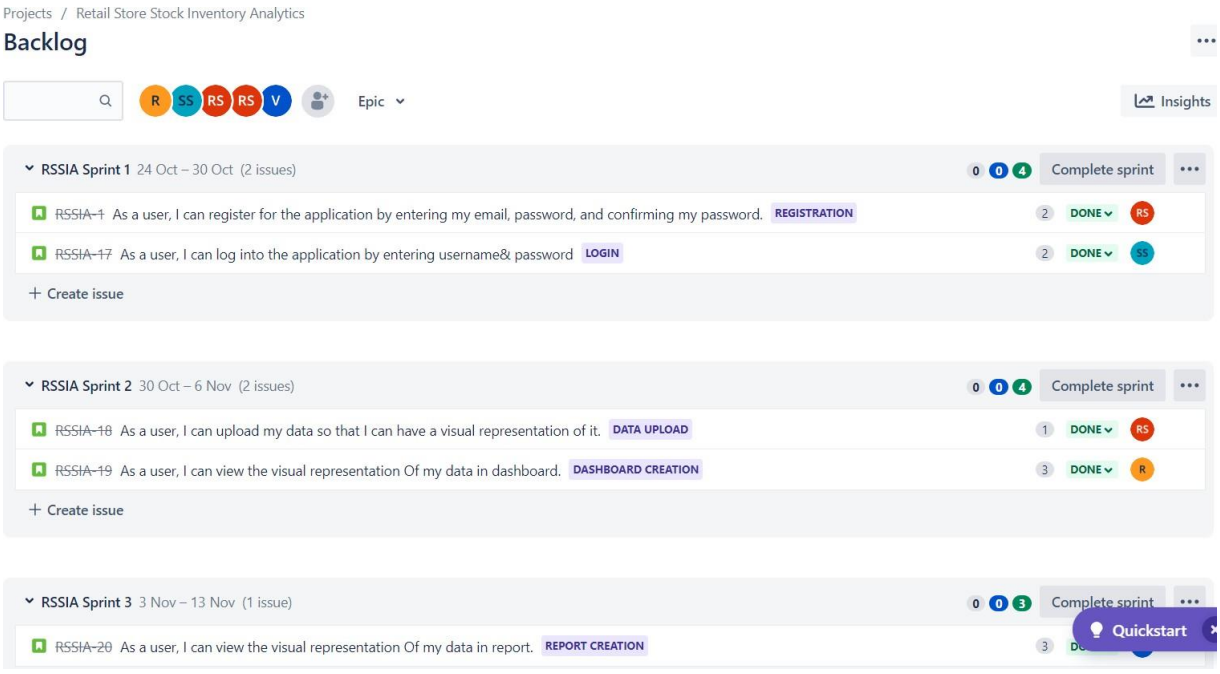
In Scrum project sprint delivery schedule is used to estimate when sprint has started and delivery date of the sprint.Due to estimation of the sprint delivery schedule it helps the developer to complete their project within the estimated time.

Reports from JIRA:

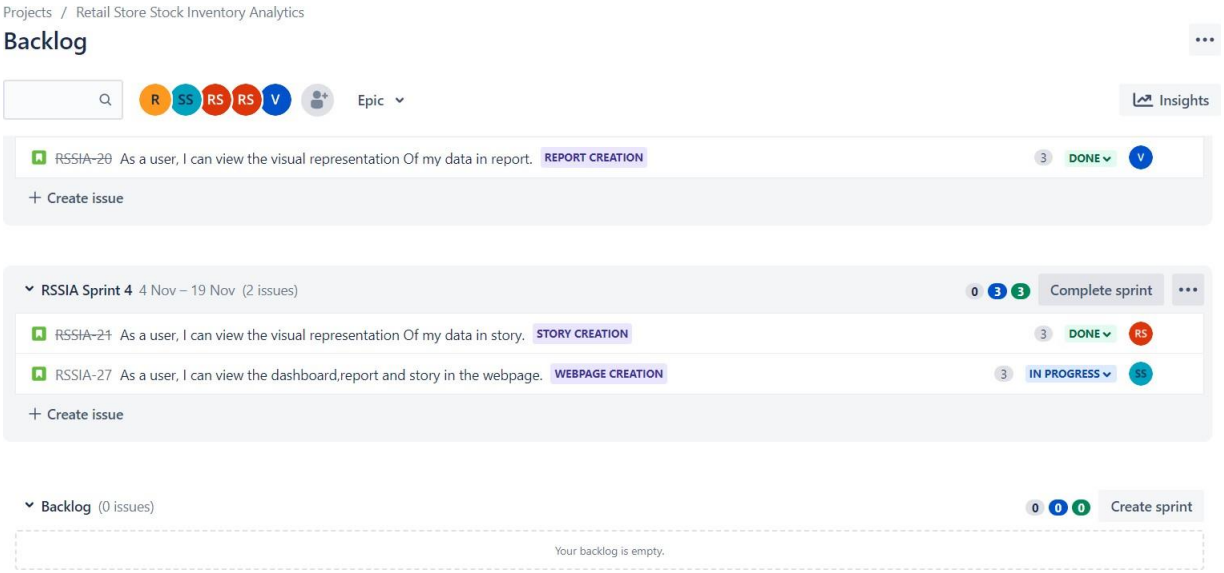
The reports in jira has been denoted below:

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).

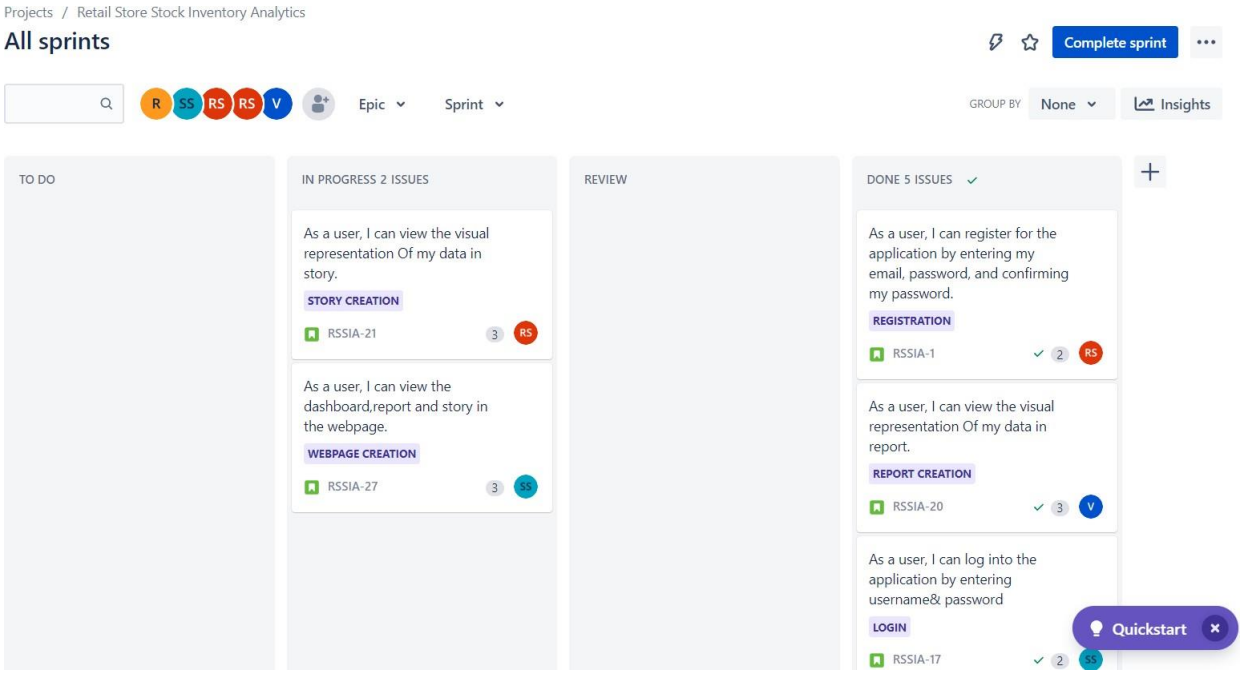


Retail Store Stock Inventory Analytics

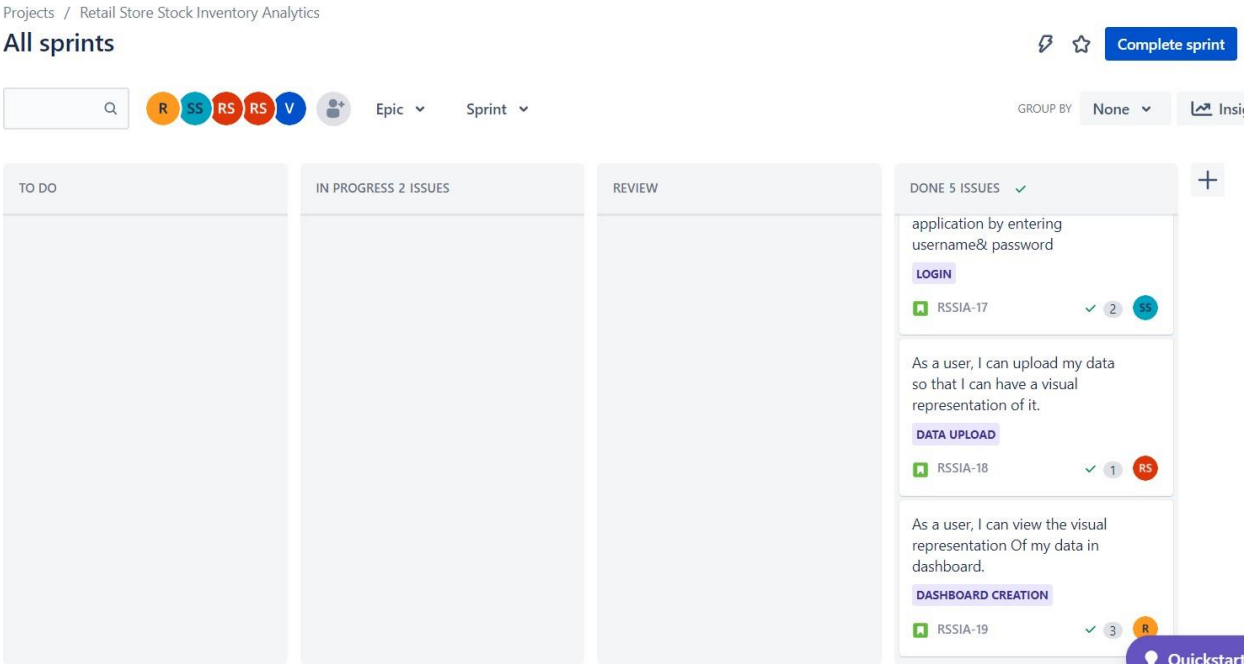


BOARD:

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.

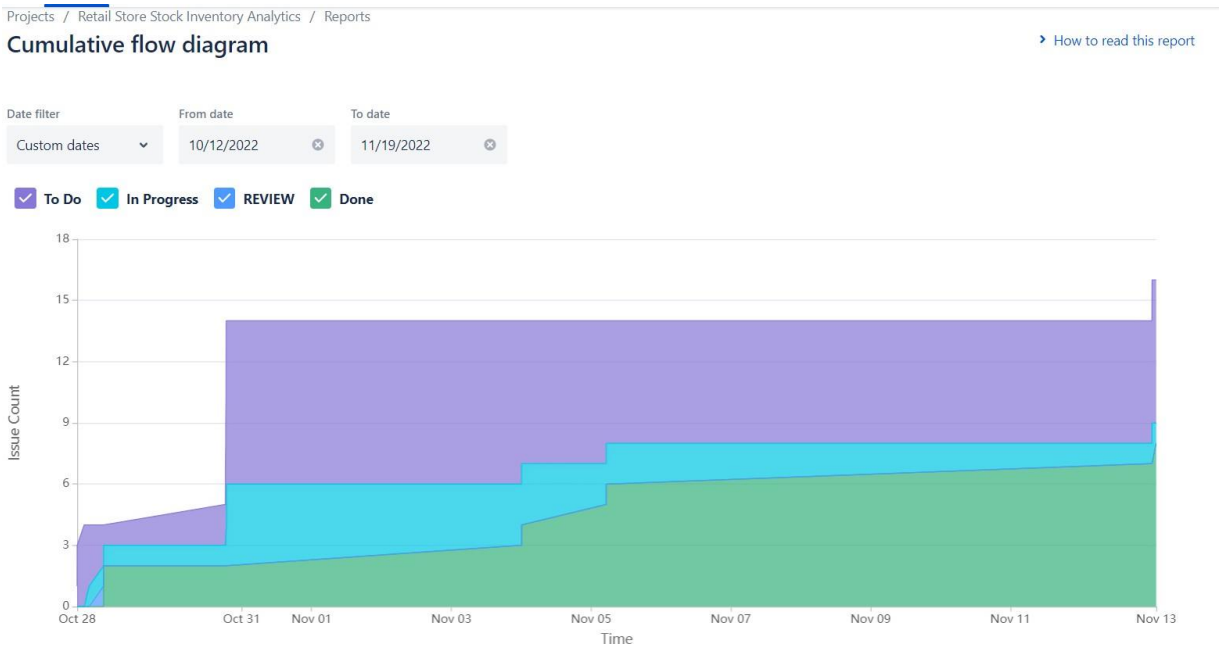


Retail Store Stock Inventory Analytics



COMMULATIVE FLOW DIAGRAM:

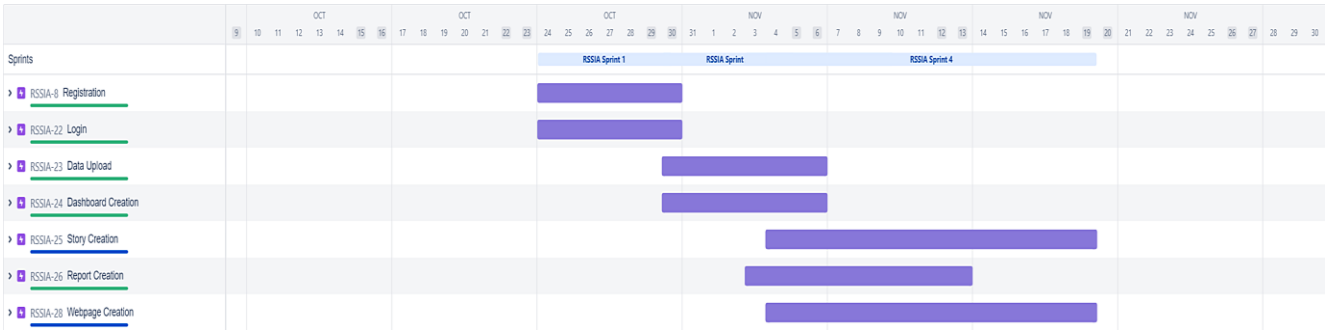
A Cumulative Flow Diagram (CFD) is an area chart that shows the various statuses of work items for an application, version, or sprint. The horizontal x-axis in a CFD indicates time, and the vertical y-axis indicates cards (issues).



Retail Store Stock Inventory Analytics

ROAD MAP:

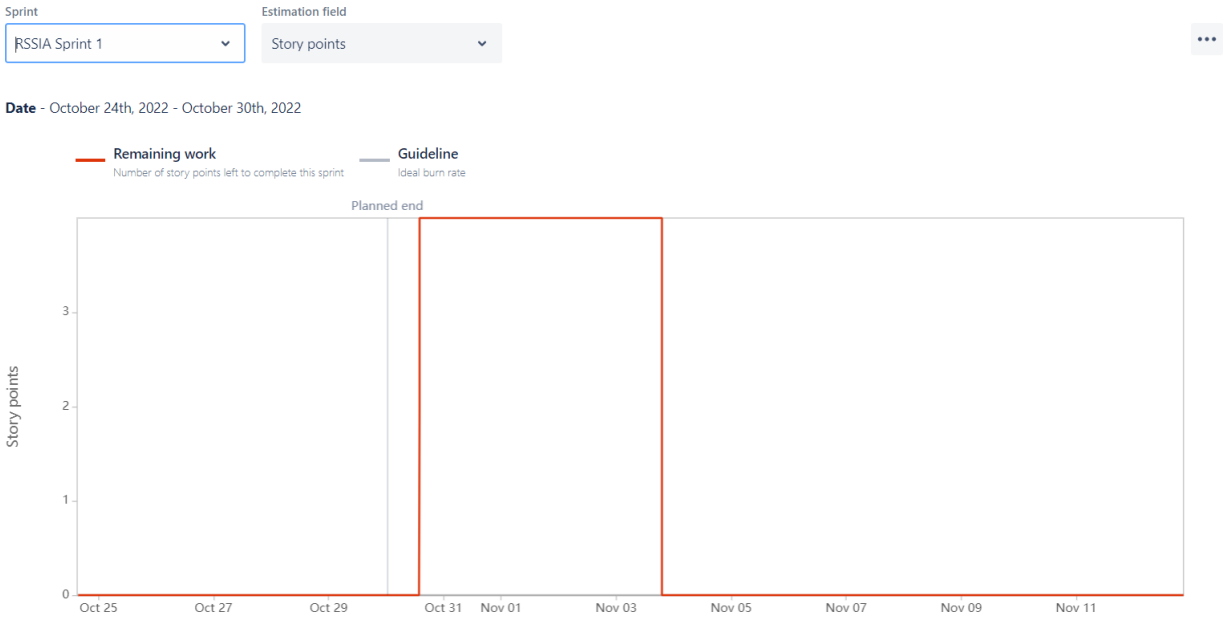
Roadmaps in Jira Software are team-level roadmaps useful for planning large pieces of work several months in advance at the Epic level within a single project. Simple planning and dependency management features help your teams visualize and manage work better together.



SPRINT BURNDOWN CHART:

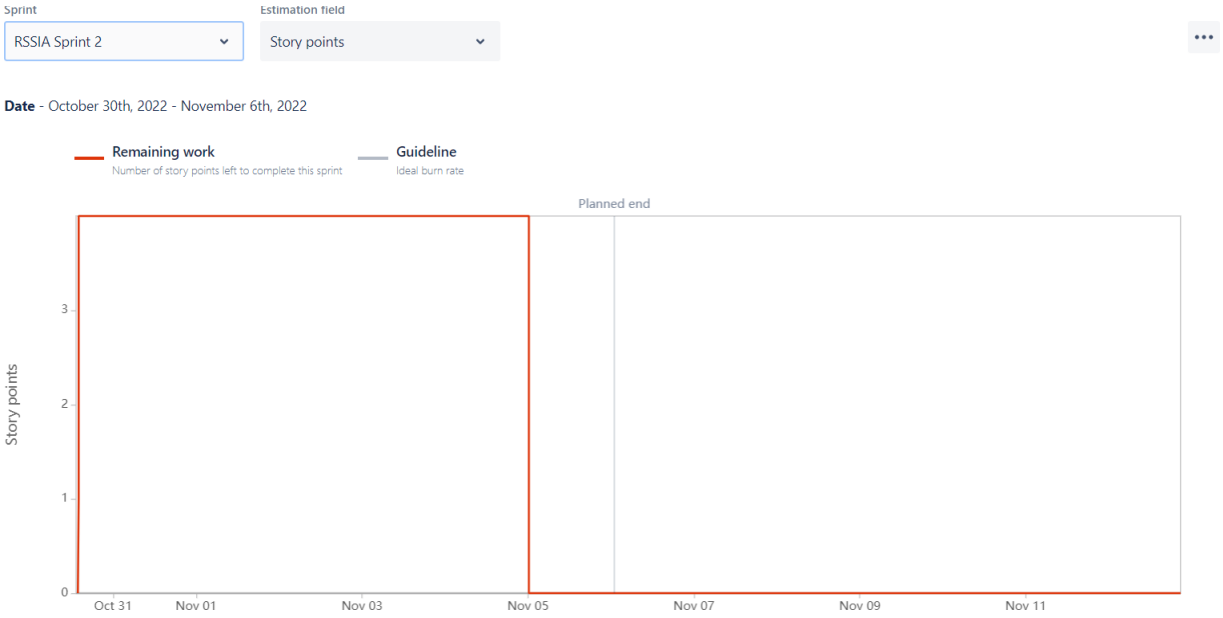
A burndown chart shows the amount of work that has been completed in an epic or sprint, and the total work remaining. Burndown charts are usedto predict your team's likelihood of completing their work in the time available.

SPRINT1:

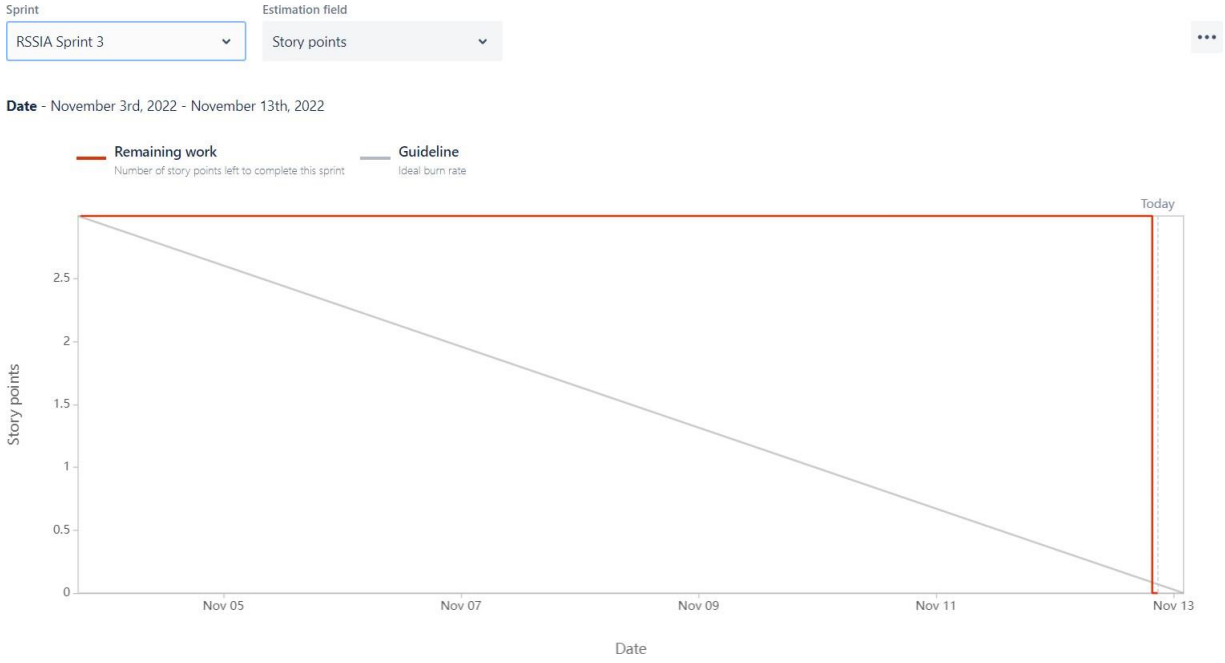


SPRINT2:

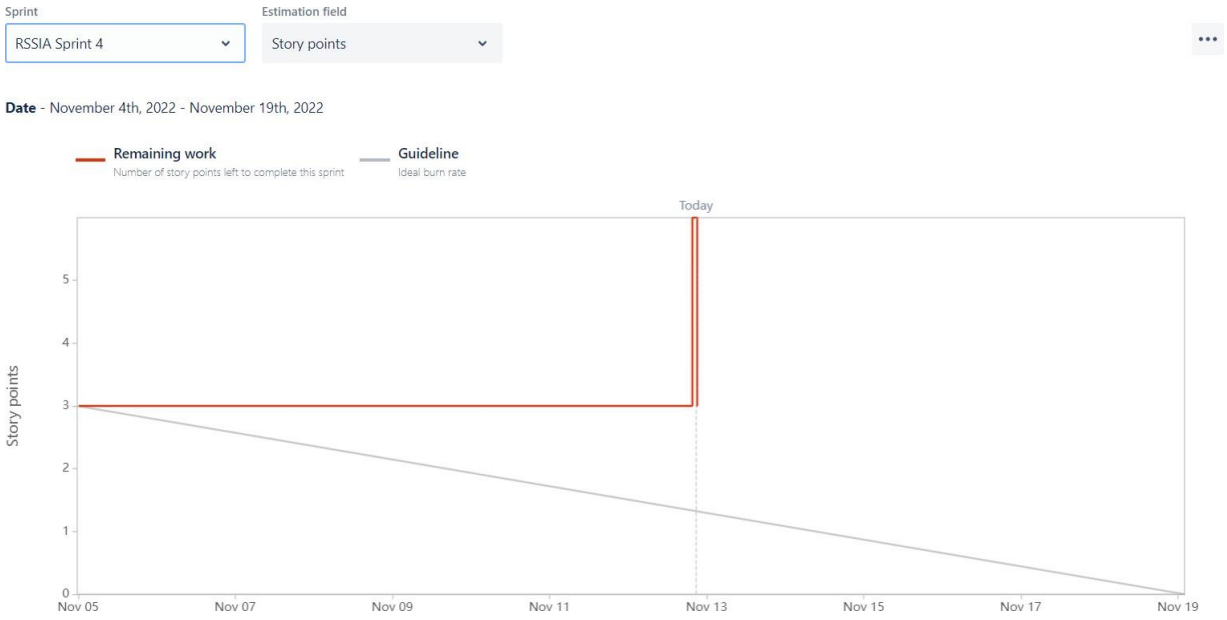
Retail Store Stock Inventory Analytics



SPRINT3:



SPRINT4:



7. CODING & SOLUTIONING

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
use Phppot\Member;
```

```
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.mi
n.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-TIBVftvqo=");
      background-position: center; /* Center the image */
  background-repeat: no-repeat; /* Do not repeat the image */
  background-size: cover; /* Resize the background image to cover theentire
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">
```

```
                                        <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>
```

```
	<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>
```

Feature 2: Registration

- 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
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 theentire
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
   if (! empty($registrationResponse["status"])) {
      ?>
            <?php
      if ($registrationResponse["status"] == "error") {
         ?>
                        <div class="server-response error-msg"><?phpecho
$registrationResponse["message"]; ?></div>
            <?php
      } else if ($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>
```

Feature 3: Display of Dashboard, Report and Story

- 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

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

Code:

```
<?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.mi
n.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 theentire
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">
     <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>
```

Database Schema :MySQL

- 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";
```

```
/*!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

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.

Pre-Requisite	Steps To Execute	Test Data
Checks whether the logged in username is registered in backend.	1.Enter your username 2.Enter your password 3.click Login now button	username: Amirtha password:amirtha@2812
Checks whether the logged in username is not registered in backend.	1.Enter your username 2.Enter your password 3.click Login now button	username:Amirtha password:amirtha@2812
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
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:Amirtha password:amirtha@2812
checks whether the user name is present in the database.	1.Enter your username 2.Enter your password 3.click Login now button	username:raja password:abcd

Test case ID	Feature Type	Component	Test Scenario
Testcase_1	Functional	Login Page	Verifies whether the user can login if he/she was an registered user
Testcase_2	Functional	Login Page	Verifies whether an unregistered user cannot proceed with the login.
testcase_3	Functional	register page	Verifies whether an unregistered user can successfully register as an user.
testcase_4	Functional	Register page	Verifies whether an register user cannot register themself as an new user.
Testcase_5	Functional	Login page	Verifies whether an alert message popsup when an unregistered user tries to login .

Expected Result	Actual Result	Status
Homepage should display	Working as expected	Pass
Homepage will not display	Working as expected	pass
User will be able to access to login page	working as expected	pass
User will not be able to access to login page	working as expected	pass
message should display	Working as expected	Pass

Retail Store Stock Inventory Analytics

User Acceptance Testing

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.

Retail Store Stock Inventory Analytics

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

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
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/K68vbdEjh
4u" crossorigin="anonymous">
<style>
   body{
      background-image:url("https://images.pexels.com/photos/5011647/pexels-
photo-5011647.jpeg?auto=compress&cs=tinysrgb&w=1260&h=750&dpr=2");
      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 = "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="report1.html">Report 1</a>
     <a href="report2.html">Report 2</a>
  </div>
</div>
    <a href = "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, #6699CC
 , 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
require_once___DIR___. "/login.php";
```

3) Login.php

```
    <?php
use Phppot\Member;

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/K68vbdEjh4
u" crossorigin="anonymous">
<style>
        body{
       background-image:url("https://media.istockphoto.com/photos/futuristic-
digital-block-chain-background-picture-
id1212911887?b=1&k=20&m=1212911887&s=612x612&w=0&h=gXEH3M0b4zMv
QzONRaU13ErOR6bgV0BCa101sklr27Y=");
       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, #808080, #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">
                            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>

                    <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>
```

4)logout.php

```
<?php
session_start();
session_unset();
session_write_close();
$url = "./index.php";
header("Location:$url");
```

5)User-registration.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://img.freepik.com/premium-
vector/geometric-hi-tech-background_29971-442.jpg?w=900");
    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,#00008B
,#808080);
}
.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
   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
/**
 * Copyright (C) Phppot
 *
 * Distributed under 'The MIT License (MIT)'
 * In essense, you can do commercial use, modify, distribute and private use.
 * Though not mandatory, you are requested to attribute Phppot URL in your code or website.
 */
namespace Phppot;

/**
 * Generic datasource class for handling DB operations.
 * Uses MySqli and PreparedStatements.
 *
 * @version 2.7 - PDO connection option added
 */
class DataSource
{

    // PHP 7.1.0 visibility modifiers are allowed for class constants.
    // when using above 7.1.0, declare the below constants as private
    // for better encapsulation
    const HOST = 'localhost';

    const USERNAME = 'root';

    const PASSWORD = '';

    const DATABASENAME = 'signup';

    private $conn;

    /**
```

```
     * PHP implicitly takes care of cleanup for default connection types.
     * So no need to worry about closing the connection.
     *
     * Singletons not required in PHP as there is no
     * concept of shared memory.
     * Every object lives only for a request.
     *
     * Keeping things simple and that works!
     */
    function __construct()
    {
        $this->conn = $this->getConnection();
    }

    /**
     * If connection object is needed use this method and get access to it.
     * Otherwise, use the below methods for insert / update / etc.
     *
     * @return \mysqli
     */
    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;
    }

    /**
     * If you wish to use PDO use this function to get a connection instance
```

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

/**
 * To get database results
 *
 * @param string $query
 * @param string $paramType
 * @param array $paramArray
 * @return array
 */
public function select($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;
    }
}

/**
 * To insert
 *
 * @param string $query
 * @param string $paramType
 * @param array $paramArray
 * @return int
 */
public function insert($query, $paramType, $paramArray)
{
    $stmt = $this->conn->prepare($query);
    $this->bindQueryParams($stmt, $paramType, $paramArray);

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

/**
 * To execute query
 *
 * @param string $query
```

```
 * @param string $paramType
 * @param array $paramArray
 */
public function execute($query, $paramType = "", $paramArray = array())
{
    $stmt = $this->conn->prepare($query);

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

/**
 * 1.
 * Prepares parameter binding
 * 2. Bind prameters to the sql statement
 *
 * @param string $stmt
 * @param string $paramType
 * @param array $paramArray
 */
public function bindQueryParams($stmt, $paramType, $paramArray = array())
{
    $paramValueReference[] = & $paramType;
    for ($i = 0; $i < count($paramArray); $i ++) {
        $paramValueReference[] = & $paramArray[$i];
    }
    call_user_func_array(array(
        $stmt,
        'bind_param'
    ), $paramValueReference);
}
```

```
    /**
     * To get database results
     *
     * @param string $query
     * @param string $paramType
     * @param array $paramArray
     * @return array
     */
    public function getRecordCount($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
namespace Phppot;

class Member
{

    private $ds;

    function __construct()
    {
        require_once __DIR__ . '/../lib/DataSource.php';
        $this->ds = new DataSource();
    }
```

```
/**
 * to check if the username already exists
 *
 * @param string $username
 * @return boolean
 */
public function isUsernameExists($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;
}

/**
 * to check if the email already exists
 *
 * @param string $email
 * @return boolean
 */
public function isEmailExists($email)
{
    $query = 'SELECT * FROM tbl_member where email = ?';
    $paramType = 's';
    $paramValue = array(
```

```
        $email
    );
    $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;
}

/**
 * to signup / register a user
 *
 * @return string[] registration status message
 */
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'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(
        $username
    );
    $memberRecord = $this->ds->select($query, $paramType, $paramValue);
    return $memberRecord;
}

/**
 * to login a user
 *
 * @return string
```

```
     */
    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) 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 {
	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"] {
	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;
}

.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}

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

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

9) dashboard.html

```
<head>
<bod<html>
y>
<img src="d.jpg">
</head>
      </html>
```

10) story.html

```
<html>
<head>
```

```
<body>
<img src="s.jpg">
</head>
    </html>
```

11) report1.html

```
<html>
<head>
    <style>
    body{
    padding: 30px;
    }
</style>
</head>
<body>
    <div class="wrapper">
    <div class="container">
     <div class="row">
          <div class="col-4">
               <img image-resize src="4.png">
          </div>
          <div class="col-4">
               <img image-resize src="5.png">
          </div>

    </html>
```

12) report2.html

```
<html>
<head>
<body>
<img src="1.png">
```

```
<img src="2.png">
<img src="3.png">
</p>
</head>
    </html>
```

GitHub & Project Link:

GitHub Link: https://github.com/IBM-EPBL/IBM-Project-54031-1661587816