

FARMCONNECT

Introduction

FarmConnect is a web platform built to bring farmers and consumers together, removing middlemen and creating a direct connection between them. This allows farmers to take charge of their product pricing, leading to better profits and more independence. One of the platform's standout features is the bargaining tool, which makes price negotiations open and fair for everyone involved. We also put a high priority on keeping user data secure, using Java for backend development and Firebase for real-time, secure data management.

Software and Hardware Requirements

Software:

- **JDK (Java Development Kit):** Powers the backend logic and ensures smooth database connectivity.
- **IDE (Integrated Development Environment) :**NetBeans is used for development.
- **HTML, CSS, JavaScript:** Used to design and style the frontend for an interactive, user-friendly experience.
- **Bootstrap:** A framework that helps create responsive, mobile-friendly web pages.
- **MySQL:** A relational database for storing user data, transactions, and product information.
- **JSP (JavaServer Pages):** Dynamically generates web pages and handles user interactions.
- **JSTL and EL:** These technologies simplify the management of dynamic content on JSP pages.
- **Apache Tomcat:** A web server to run and manage the website.

Hardware:

- **Server:** Either cloud-based or on-premise, hosting the website and handling all traffic.
- **Development Machine:** Used for building and testing the website.
- **Database Server:** Hosts MySQL or another database to handle all user and transaction data.

Implementation

1. Project Setup:

The website is built using JDK and an IDE NetBeans. We focus on modularity to make the platform easy to expand and maintain.

2. Database Design:

A clear and efficient MySQL schema stores user details, products, transactions, and price negotiation data. Tables are created to manage all this information seamlessly.

3. Database Connectivity with JDBC:

JDBC connects the backend to the MySQL database, making sure that data like user information and transactions flow smoothly between the frontend and backend.

4. DAO Classes:

These classes are responsible for all interactions with the database, ensuring smooth operations like creating, reading, updating, and deleting user and product data.

5. Frontend Development:

HTML templates are used to build the structure of the website, while CSS and Bootstrap style the pages. JavaScript brings it all together, providing interactive features and form validation.

6. Form Validation with JavaScript:

JavaScript helps ensure that users enter the correct data into forms, minimizing errors before submitting information.

7. Servlet Configuration:

Servlets handle user requests and responses, processing tasks like registration, login, and transactions, while passing data between the frontend and backend.

8. doGet and doPost Methods:

These methods manage the retrieval and submission of data on the website. For instance, the doGet method retrieves data, and the doPost method handles updates or new submissions like user registration or price negotiations.

9. User Registration and Profile Management:

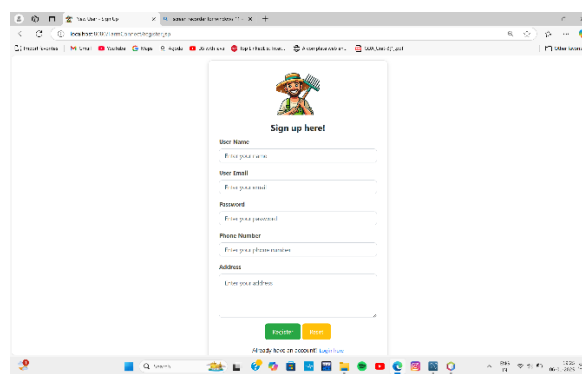
The platform allows users to sign up, manage their profiles, and control their products and pricing directly.

10. JSP Integration:

JSP is used to dynamically show user data like product listings, profiles, and transaction history. JSTL and EL help manage the content easily, making sure it displays in an organized way.

Project Prototype

Registration Page

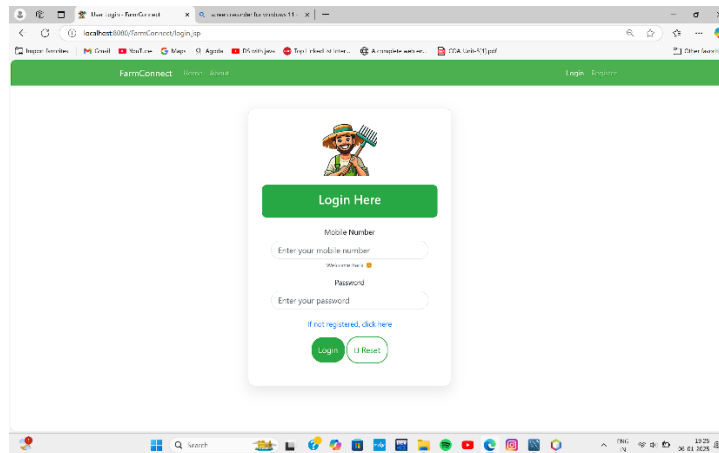


The screenshot displays a web browser window with a registration form. The form is titled "Sign up here!" and features a cartoon character icon. It contains the following fields and labels:

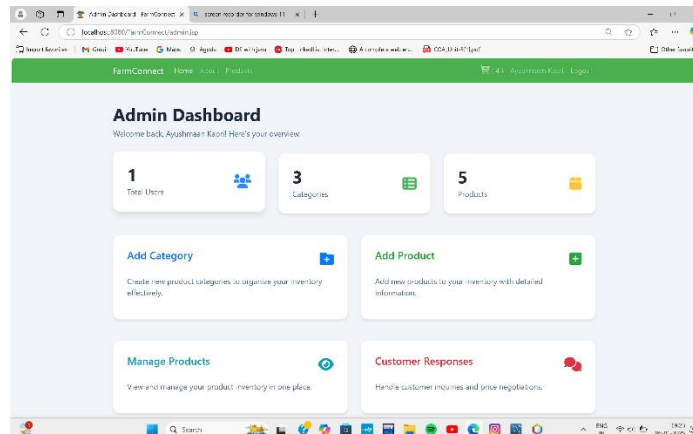
- User Name: Enter your name
- User Email: Enter your email
- Password: Enter your password
- Phone Number: Enter your phone number
- Address: Enter your address

At the bottom of the form, there are two buttons: "register" (green) and "login" (yellow). The browser's address bar shows the URL "http://localhost:8080/".

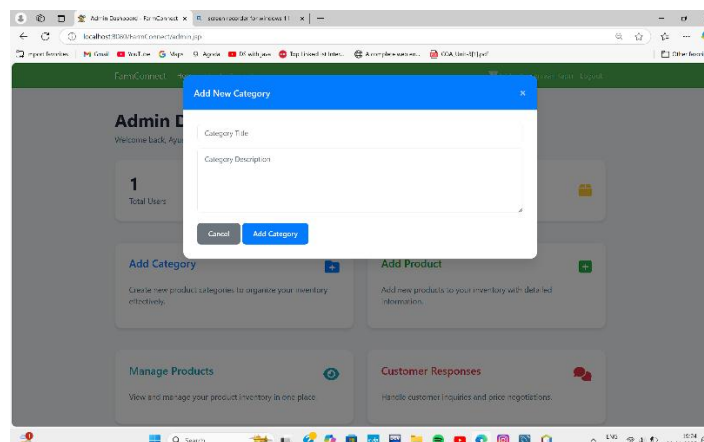
Login Page



Admin Page



Add Category Page



Add Product

The screenshot shows the 'Add New Product' modal form in the FarmConnect Admin Dashboard. The form is a white box with a green header and a green 'Add Product' button. It contains the following fields:

- Product Title
- Product Description
- Price
- Discount %
- Quantity
- Vendor/Supplier
- Product Image
- Choose File: No file chosen

The background shows the Admin Dashboard with a sidebar and a main content area.

Manage Products

The screenshot shows the 'Manage Your Products' page in the FarmConnect Admin Dashboard. It features a table with the following columns: ID, Name, Description, Price, and Action. The table contains 5 rows of product data.

ID	Name	Description	Price	Action
1	Tomatoes	Fresh red tomatoes	₹ 54	Edit Delete
2	Apples	Red Apples	₹ 72	Edit Delete
3	Basmati Rice	Best Quality Basmati Rice	₹ 120	Edit Delete
4	Red Capsicum	Fresh Red Capsicum	₹ 45	Edit Delete
5	Bananas	Fresh Bananas	₹ 54	Edit Delete

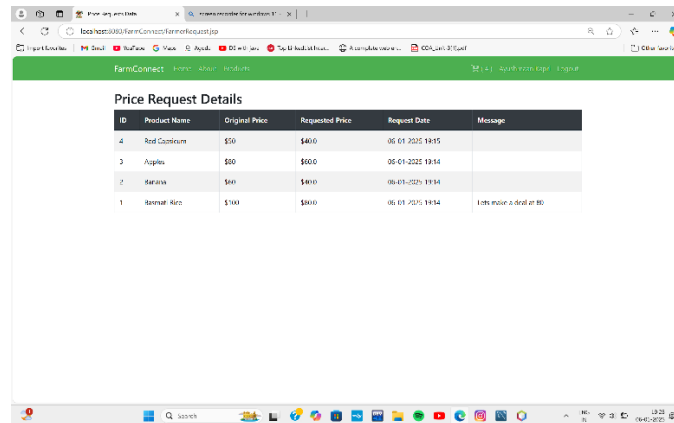
Edit Products

The screenshot shows the 'Edit Product' form in the FarmConnect Admin Dashboard. The form is a white box with a green header and a green 'Update Product' button. It contains the following fields:

- Product Name
- Description
- Price
- Discount (%)
- Quantity
- Current Image
- Update Image (optional)
- Choose File: No file chosen

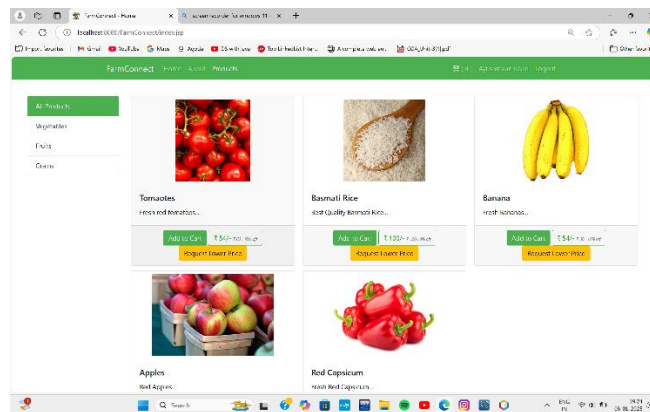
The background shows the Admin Dashboard with a sidebar and a main content area.

Customer Responses

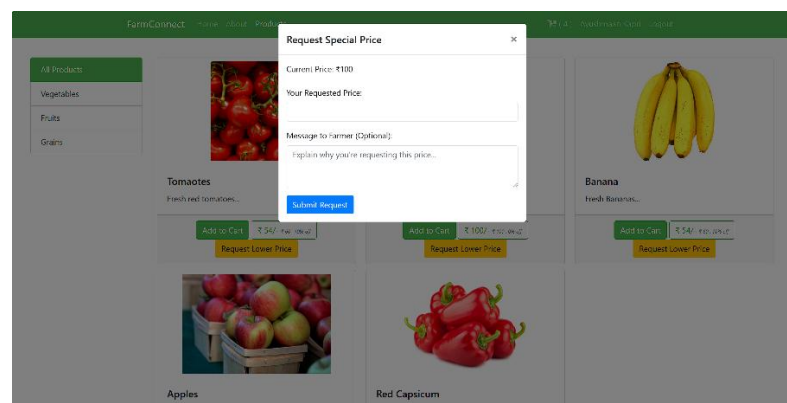


ID	Product Name	Original Price	Requested Price	Request Date	Message
4	Red Capsicum	\$50	\$40.0	06-01-2025 19:55	
3	Apples	\$80	\$60.0	06-01-2025 19:54	
2	Bananas	\$80	\$60.0	06-01-2025 19:54	
1	Basmati Rice	\$100	\$80.0	06-01-2025 19:54	I like more a deal at 80

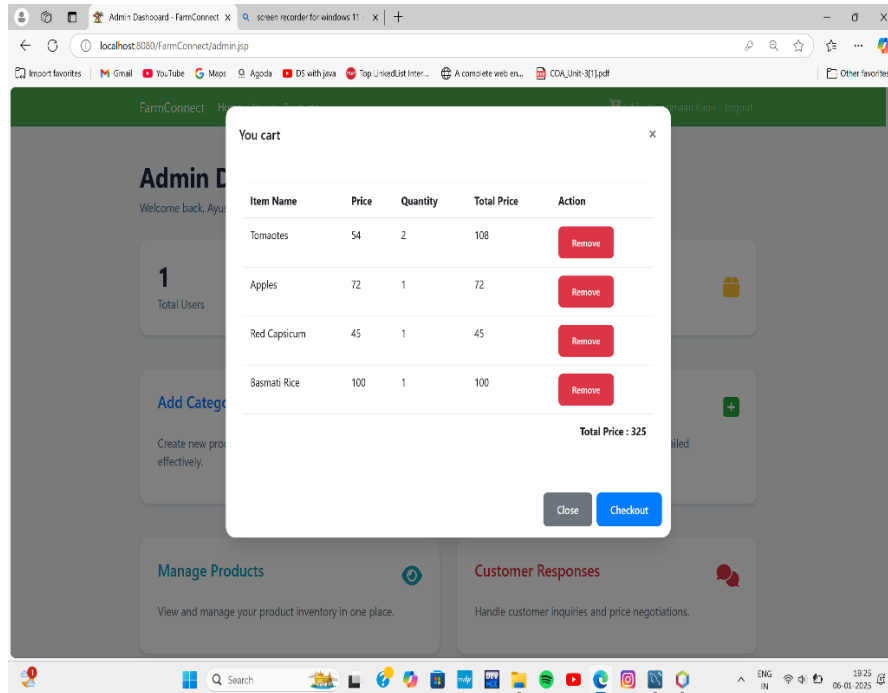
Products (Customer View)



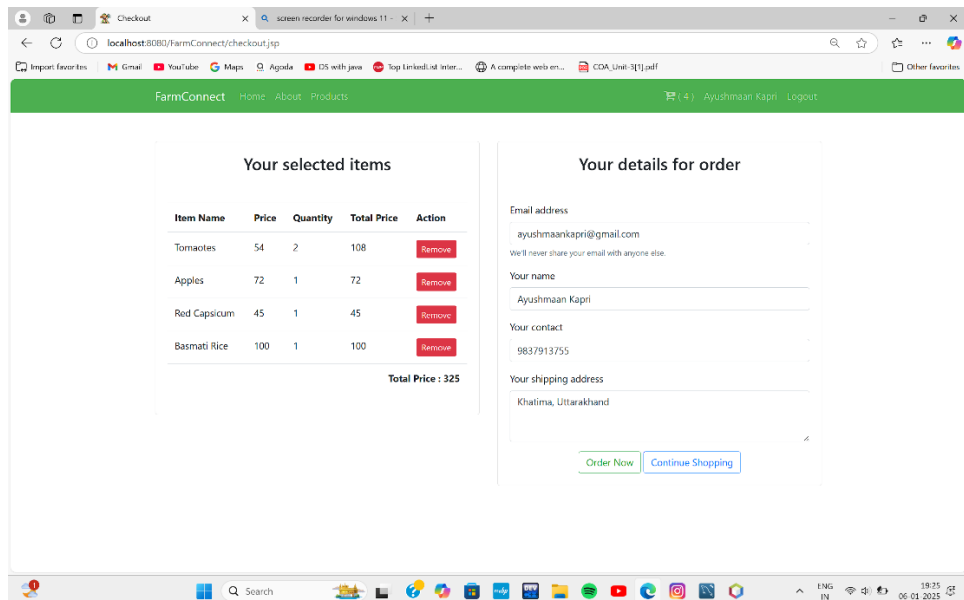
Request Special Price



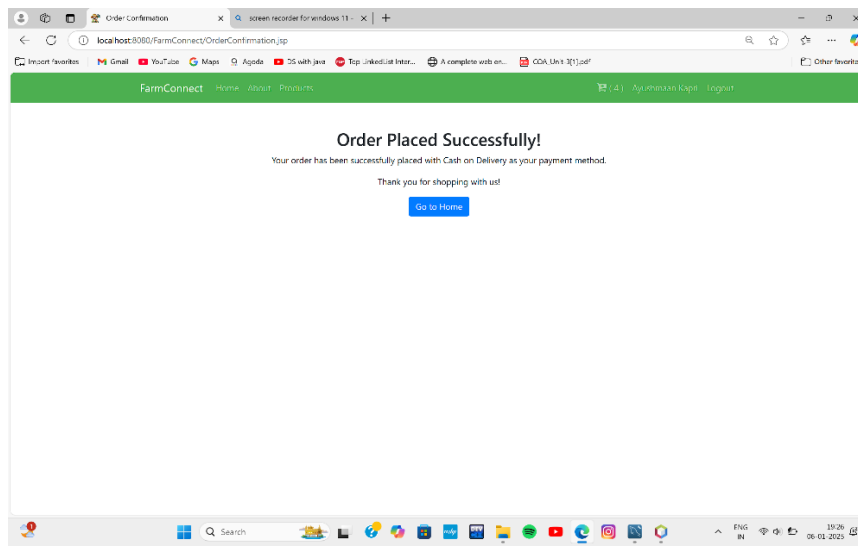
Cart



Checkout Page

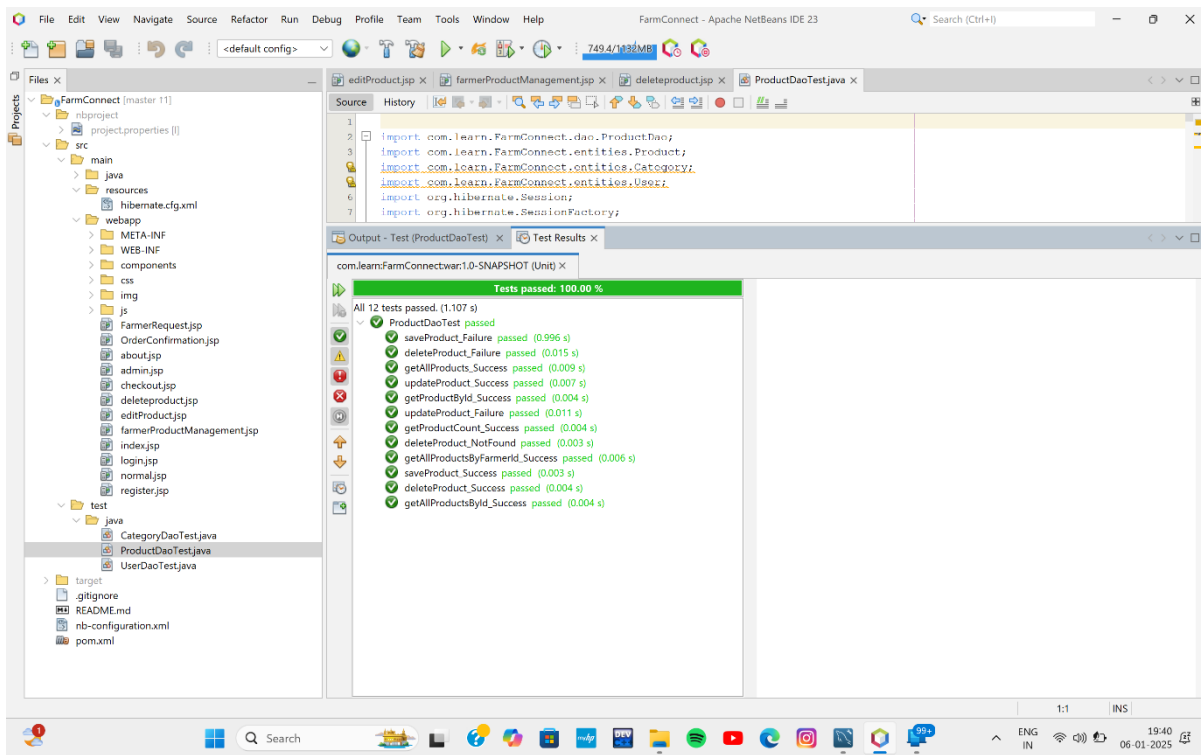


Order Confirmation

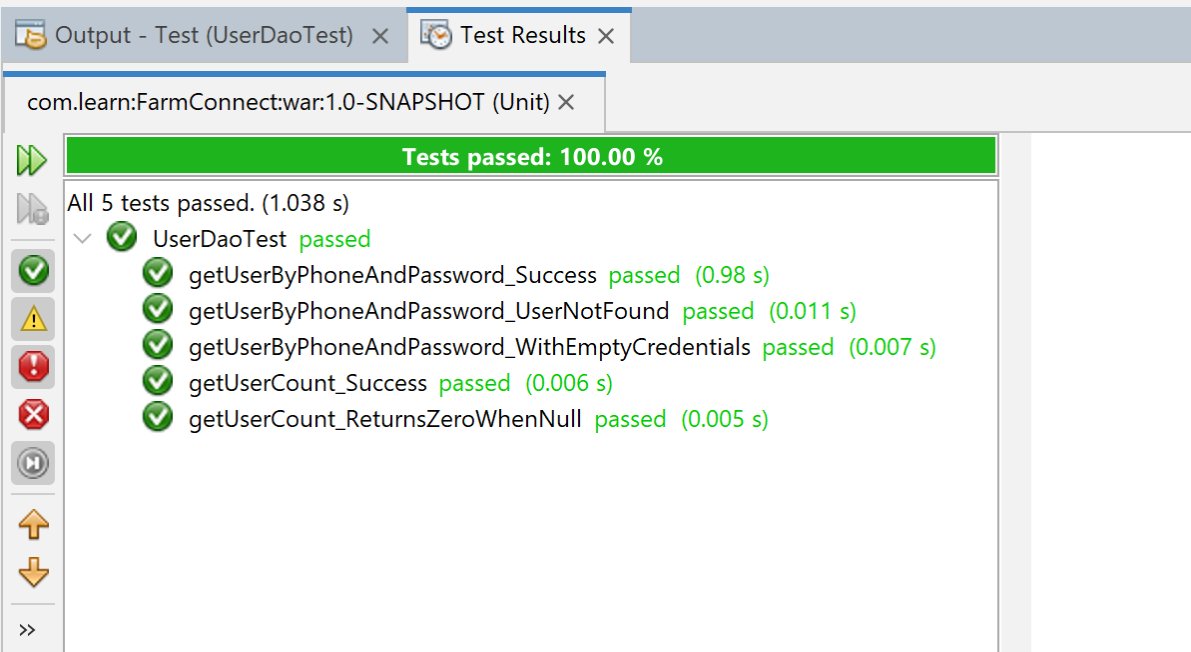


Test Pages

Product Dao Test



User Dao Test

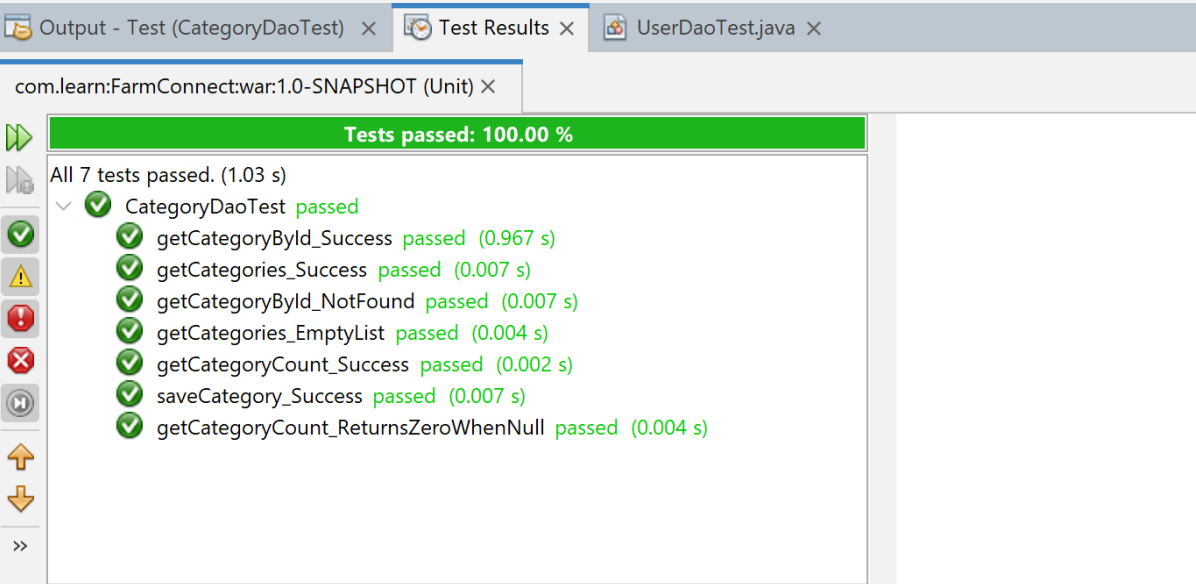


The screenshot shows the IntelliJ IDEA Test Results window for the `UserDaoTest` class. The window has two tabs: "Output - Test (UserDaoTest)" and "Test Results". The "Test Results" tab is active, showing a green bar at the top indicating "Tests passed: 100.00 %". Below this, it states "All 5 tests passed. (1.038 s)". A tree view shows the test results for `UserDaoTest`, which is marked as "passed". Under `UserDaoTest`, there are five sub-tests, all marked with green checkmarks and "passed":

- `getUserByPhoneAndPassword_Success` passed (0.98 s)
- `getUserByPhoneAndPassword_UserNotFound` passed (0.011 s)
- `getUserByPhoneAndPassword_WithEmptyCredentials` passed (0.007 s)
- `getUserCount_Success` passed (0.006 s)
- `getUserCount_ReturnsZeroWhenNull` passed (0.005 s)

On the left side of the window, there is a vertical toolbar with icons for running tests, showing test results, and navigating between tests.

Category Dao Test



The screenshot shows the IntelliJ IDEA Test Results window for the `CategoryDaoTest` class. The window has three tabs: "Output - Test (CategoryDaoTest)", "Test Results", and "UserDaoTest.java". The "Test Results" tab is active, showing a green bar at the top indicating "Tests passed: 100.00 %". Below this, it states "All 7 tests passed. (1.03 s)". A tree view shows the test results for `CategoryDaoTest`, which is marked as "passed". Under `CategoryDaoTest`, there are seven sub-tests, all marked with green checkmarks and "passed":

- `getCategoryById_Success` passed (0.967 s)
- `getCategories_Success` passed (0.007 s)
- `getCategoryById_NotFound` passed (0.007 s)
- `getCategories_EmptyList` passed (0.004 s)
- `getCategoryCount_Success` passed (0.002 s)
- `saveCategory_Success` passed (0.007 s)
- `getCategoryCount_ReturnsZeroWhenNull` passed (0.004 s)

On the left side of the window, there is a vertical toolbar with icons for running tests, showing test results, and navigating between tests.

Results

- **Empowered Farmers:**
By removing middlemen, farmers now have full control over the prices of their products, leading to increased profits.
- **Transparent and Fair Pricing:**
The bargaining feature gives both farmers and consumers the opportunity to negotiate prices fairly, ensuring everyone is satisfied.
- **Efficient Data Handling:**
Firebase ensures that data is securely stored and updated in real-time, while JDBC helps facilitate smooth communication between the website and database.
- **User-Friendly Interface:**
Built with HTML, CSS, JavaScript, and Bootstrap, the platform is designed to work seamlessly on any device, providing a great experience for every user.
- **Secure Transactions:**
Transactions are encrypted and secure, ensuring that all user data, especially payment details, are protected.

Conclusion

FarmConnect is more than just a website; it's a solution to the challenges farmers face in reaching consumers and getting fair prices for their products. By cutting out the middlemen, the platform empowers farmers to control their pricing and connect directly with consumers. With a focus on security, responsiveness, and simplicity, FarmConnect is setting the stage for a more sustainable and fair agricultural marketplace. Using technologies like, JDBC, JSP, and Bootstrap, we've built a platform that's secure, scalable, and ready to meet the needs of both farmers and consumers in a digital world.