EXP. NO: 10

IMPLEMENTING AND TESTING THE MODIFIED SYSTEM

Aim:

To implement the improved system and test it.

Improvements Made:

These enhancements address functionality, scalability, and user experience, ensuring a more robust and user-friendly platform.

1. **User Interface Enhancements:**
    - Bootstrap 5 Features: Leveraged updated Bootstrap 5 utilities and components, such as grid layouts, modals, and form controls, to make the interface more intuitive and visually appealing.
    - Responsive Design Optimization: Introduced breakpoints to adjust layouts dynamically across desktops, tablets, and mobile devices, ensuring consistent usability on different screen sizes.
    - Enhanced Accessibility: Improved accessibility by adhering to WCAG standards, adding keyboard navigation support, and ensuring screen reader compatibility.
    - Animations and Micro-interactions: Subtle animations were added to buttons, modals, and notifications to improve the overall user experience without impacting performance.
2. **Authentication and Security Improvements:**
    - Session management was improved with the addition of refresh tokens, reducing vulnerability to session hijacking.
    - Role-based access control (RBAC) was added, ensuring that sensitive operations are restricted to authorized admin users.
    - Multiple methods of authentication (Google, Phone authentication) were added in addition to Email / Password authentication. This improves security by reducing account duplication and letting users use their mobile or Google account to use the stock management system.
3. **Dashboard**:
    - The dashboard was transformed into a powerful monitoring and analytics hub to provide actionable insights at a glance.
    - **Real-Time Visual Analytics:** Introduced widgets displaying live stock status, sales trends, and billing metrics, all powered by Firebase's real-time updates.
    - **Interactive Data Visualizations:** Integrated Chart.js to provide interactive pie charts, bar graphs, and line charts for data analysis. Users can hover over data points to see detailed insights or toggle chart elements to focus on specific data.

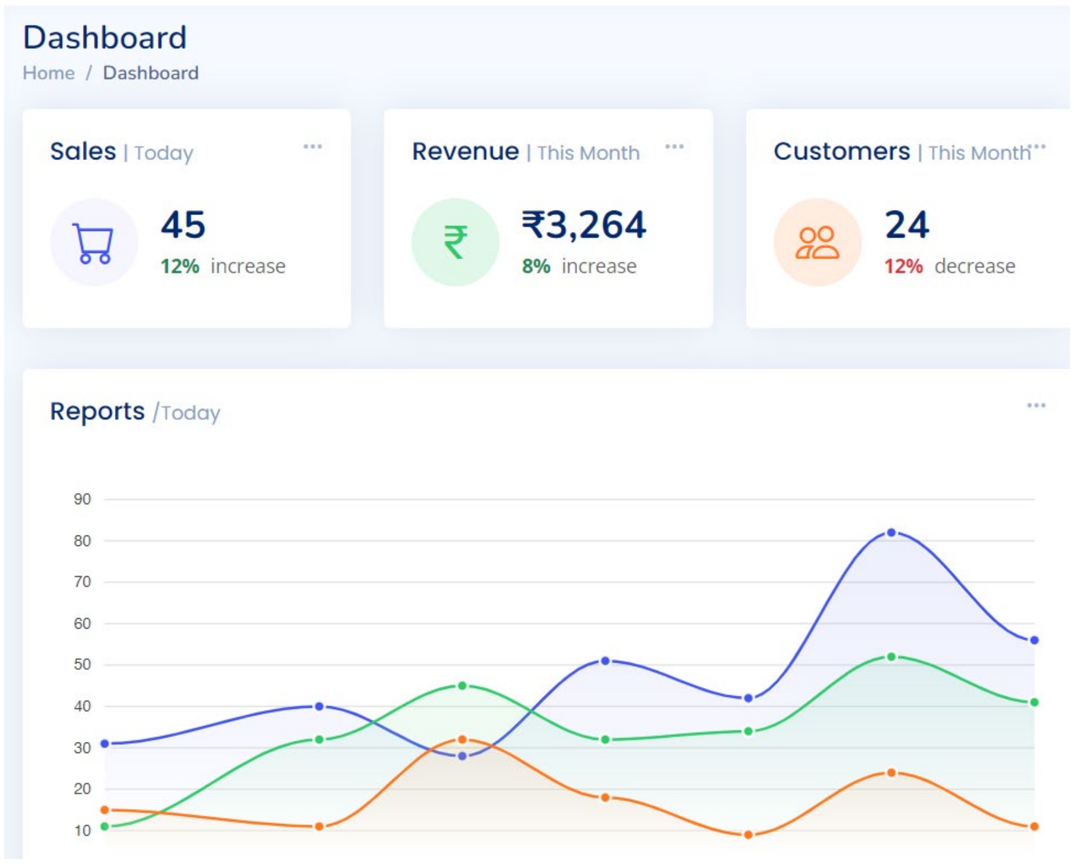
- **Downloadable Reports:** Added an export feature that allows users to download custom reports (e.g., stock summaries or monthly billing data) in PDF and Excel formats.
- **Alerts and Insights:** The dashboard now includes alert systems that notify users of critical updates, such as low stock or overdue payments.

4. **Stock Management Optimizations**
   - Introduced **batch stock operations**, allowing users to process multiple stock-in or stock-out items in a single action.
   - Real-time notifications now alert users about **low stock levels** or expired items.
   - Added a search and filter feature, enabling quick access to specific stock items by category, date, or supplier.
5. **Administrative Controls**
   - Improved user management features, including the ability to assign roles and deactivate accounts.
   - A **data backup and restore feature** was added, ensuring data resilience and recovery in case of emergencies.
   - A registration page has been added so that the Admin can register new users without having to interface with the backend directly. This makes it easy for any admin to operate the system and manage users.
6. **Performance and Scalability:**
   - The application underwent significant backend and frontend optimization to ensure faster load times and seamless scalability.
   - Firebase Cloud Functions: Heavy operations, such as generating reports and processing batch stock transactions, were moved to Firebase Cloud Functions. This reduced the client-side computation burden and improved overall responsiveness.
   - Firestore Indexing: Database queries were optimized using Firestore's compound indexes. This minimized query execution times, especially for complex filters and sorting operations.
   - Database Batching: Used batch writes for operations involving multiple database updates, reducing latency and improving data consistency.
   - Caching Mechanisms: Firebase Realtime Database listeners now cache frequently accessed data locally. This reduces redundant reads, enhances offline functionality, and speeds up data retrieval.
   - Scalable Architecture: The app's architecture was designed to handle increased traffic with minimal changes, making it ready for enterprise-level usage.

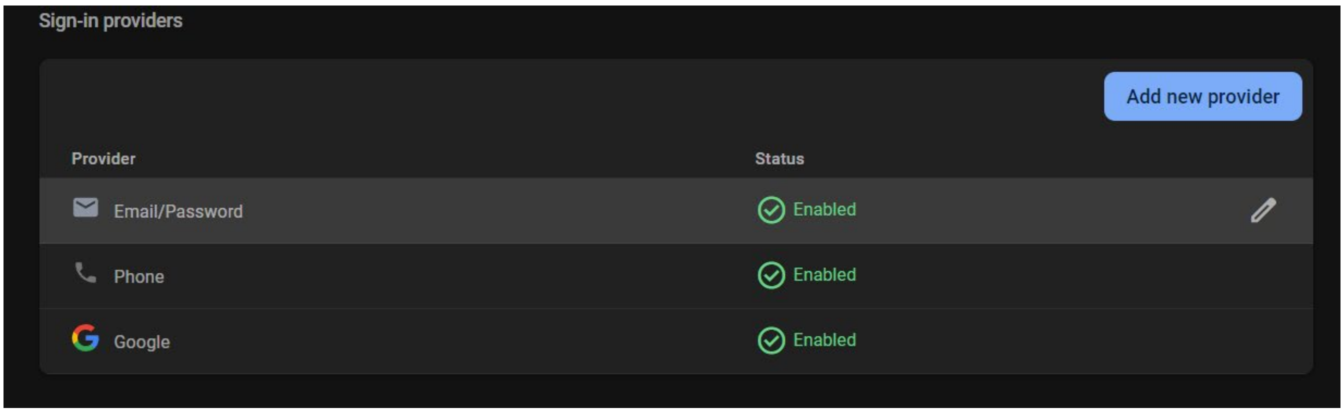
7. **Testing and Debugging Enhancements:**

- Robust testing and debugging measures were implemented to ensure reliability and ease of maintenance.
- **Automated Testing:** Added end-to-end testing scripts using Jest to validate core features, including user authentication, stock management, and billing workflows. These tests are triggered during every deployment cycle to ensure functionality remains intact.
- **Comprehensive Error Logging:** Integrated Firebase Crashlytics and logging systems to capture real-time errors and exceptions. These logs provide detailed stack traces, helping developers identify and resolve issues quickly.
- **Mock Databases for Testing:** Created mock environments with sample data to simulate real-world scenarios, allowing developers to test new features without impacting production.
- **Debugging Enhancements:** Improved error messages for users and administrators. For example, database operation errors now display user-friendly suggestions while logging technical details for developers.
- **Performance Profiling:** Leveraged tools like Firebase Performance Monitoring to analyze and fix performance bottlenecks, ensuring a smoother user experience.

Dashboard:



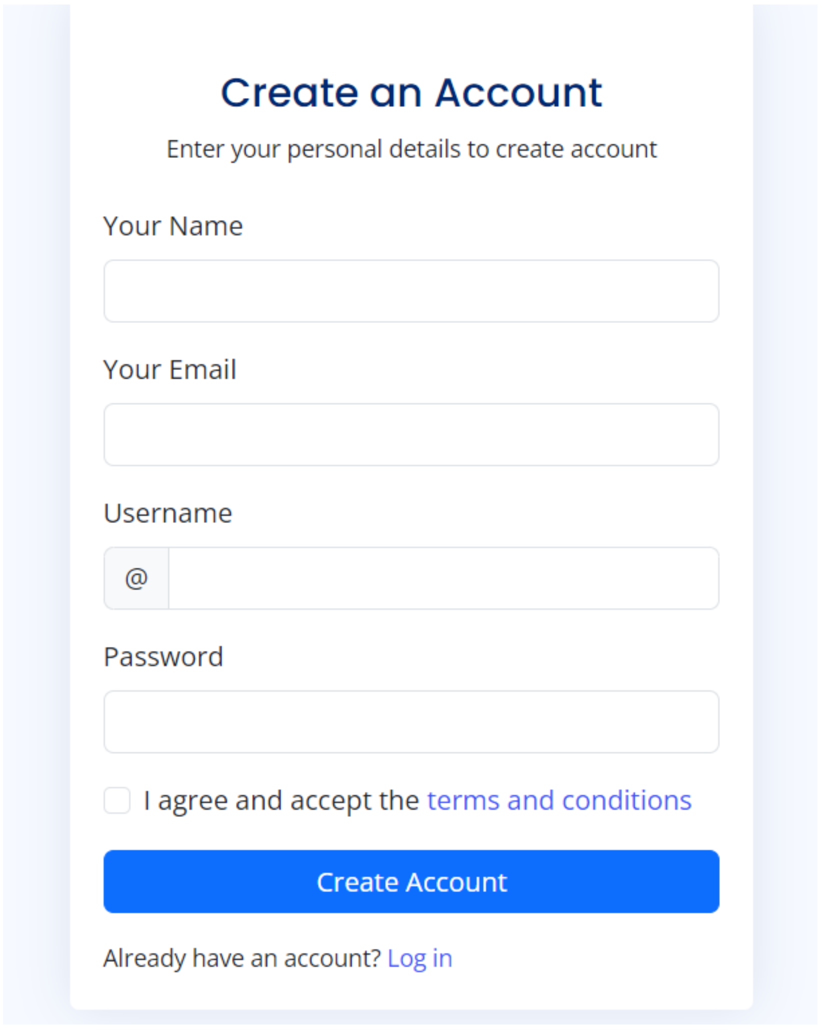
Firebase Authentication Options:


Sign-in providers

Add new provider

Provider	Status
Email/Password	Enabled
Phone	Enabled
Google	Enabled

Registration:



Create an Account

Enter your personal details to create account

Your Name

Your Email

Username

@

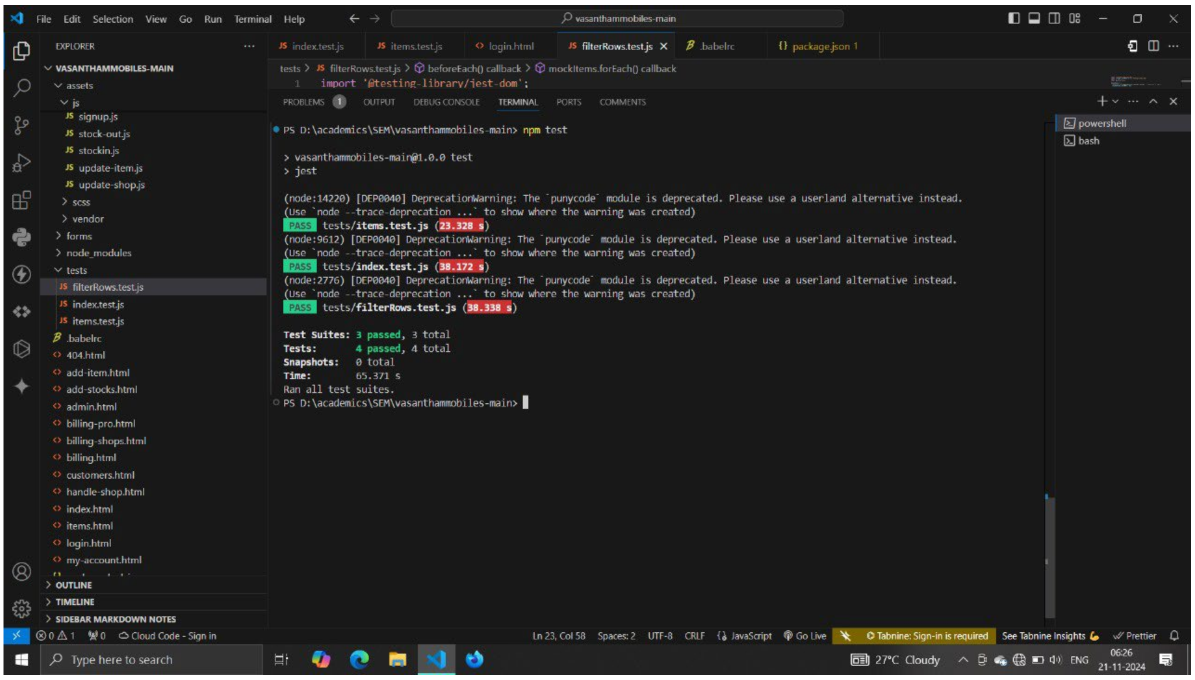
Password

I agree and accept the terms and conditions

Create Account

Already have an account? Log in

Testing with Jest:



```
PS D:\academics\SEM\vasanthammobiles-main> npm test

> vasanthammobiles-main@1.0.0 test
> jest

(node:14220) [DEP0040] DeprecationWarning: The `punycode` module is deprecated. Please use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
 PASS  tests/items.test.js (23.328 s)
(node:9612) [DEP0040] DeprecationWarning: The `punycode` module is deprecated. Please use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
 PASS  tests/index.test.js (38.172 s)
(node:2776) [DEP0040] DeprecationWarning: The `punycode` module is deprecated. Please use a userland alternative instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
 PASS  tests/filterRows.test.js (38.338 s)

Test Suites: 3 passed, 3 total
Tests:       4 passed, 4 total
Snapshots:   0 total
Time:        65.371 s
Ran all test suites.
PS D:\academics\SEM\vasanthammobiles-main>
```

Description	Allotted Marks	Obtained Marks
Preparation	**20**	
Design / Implementation	**20**	
Viva	**15**	
Output	**10**	
Record	**10**	
Total	**75**	

RESULT:

Thus, the improved Stock Inventory Management System is implemented and tested.