

📦 Stock Check Tool

A multilingual Progressive Web App (PWA) for managing daily stock inventory and restock requests via WhatsApp.

version 5.4 license ISC React 18.3 Node.js 18+

🌟 Features

Core Functionality

- **Multilingual Support** - Myanmar (မြန်မာ), English, Chinese (中文)
- **PWA Installation** - Works offline, install on any device
- **Item Selection** - Grid-based item selection with images
- **Quantity Tracking** - Individual quantities for each item
- **WhatsApp Integration** - Auto-formatted messages sent directly
- **Team Synchronization** - Real-time sync across all users
- **Unit Check** - Morning/evening quantity tracking (Braised Pork, Kong Bak)
- **Auto-Save** - Selections persist until midnight (Malaysia timezone)
- **Locked Submissions** - Prevent duplicate orders from team members

Technical Features

- ⚡ Fast, responsive UI with Tailwind CSS
- 🔒 Secure authentication with bcrypt password hashing
- 📱 Mobile-first design with touch-friendly controls
- 🔄 Automatic cache updates with version-based service worker
- 🗄 Persistent login sessions (localStorage)
- 🌐 CORS-enabled API with production deployment

🏗 Architecture

Frontend (Client)

- **Framework:** React 18.3 + Vite 7.2
- **Styling:** Tailwind CSS 4.0
- **Icons:** FontAwesome
- **Deployment:** Vercel
- **URL:** <https://client-scydom-chins-projects.vercel.app>

Backend (Server)

- **Runtime:** Node.js + Express 5.1
- **Authentication:** bcrypt password hashing
- **Database:** JSON file storage (local filesystem)
- **Deployment:** Render

- **API URL:** <https://stockcheck-api.onrender.com/api>

Database

- **users.json** - User accounts and preferences
- **submissions.json** - Daily restock submissions
- **quantities.json** - Unit check quantities (ephemeral on Render)

📁 Project Structure

```
stockchecktool/
  client/                                # Frontend React application
    public/
      sw.js          # Service worker (PWA)
      manifest.json # PWA manifest
      webp/         # Item images (WebP format)
    src/
      components/   # React components
        Header.jsx
        Tabs.jsx
        ItemGrid.jsx
        SelectedItems.jsx
        QuantitySection.jsx
        LoginForm.jsx
      context/
        ApplicationContext.jsx # Global state management
      data/
        items.js       # Item definitions (3 languages)
      utils/
        api.js        # API calls
        whatsapp.js  # WhatsApp formatting
        dateUtils.js # Malaysia timezone handling
        storage.js   # localStorage utilities
      App.jsx
      main.jsx
      App.css
    .env           # Environment variables
    package.json
    vite.config.js

  server/                                # Backend API
    src/
      database/      # JSON database files
        users.json
        submissions.json
        quantities.json
      routes/
        auth.js       # Authentication endpoints
        submissions.js # Submission endpoints
        quantities.js # Quantity check endpoints
      scripts/
```

```
|- manageUsers.js # CLI user management
|   \-- seedUsers.js # Default user seeding
|- server.js       # Express app
|- .env            # Environment variables
\-- package.json

-- DEPLOY.md          # Deployment guide
-- USER_GUIDE.md      # End-user documentation
-- OWNER_GUIDE.md     # Administrator guide
-- TODO.md            # Development checklist
-- README.md          # This file
```

🚀 Getting Started

Prerequisites

- Node.js 18+ installed
- Git installed
- npm or yarn package manager

Installation

1. Clone Repository

```
git clone https://github.com/Scydom8885/stockchecktool.git
cd stockchecktool
```

2. Setup Backend

```
cd server
npm install
```

Create `.env` file:

```
PORT=5000
NODE_ENV=development
CLIENT_URL=http://localhost:5173
JWT_SECRET=your-secret-key-change-in-production
SESSION_SECRET=your-session-secret-change-in-production
```

Seed default users:

```
npm run seed
```

Start backend:

```
npm start  
# or for development with auto-reload:  
npm run dev
```

Backend runs at: <http://localhost:5000>

3. Setup Frontend

```
cd ..../client  
npm install
```

Create `.env` file:

```
VITE_API_URL=http://localhost:5000/api
```

Start frontend:

```
npm run dev
```

Frontend runs at: <http://localhost:5173>

4. Login

Default credentials (created by seed script):

- **Username:** [admin](#)
- **Password:** [admin123](#)
- **Language:** Myanmar

🔧 CLI Commands

User Management

Navigate to `server/` directory first:

```
cd server
```

List all users:

```
npm run list-users
```

Add new user:

```
npm run add-user
```

Remove user:

```
npm run remove-user
```

Seed default users:

```
npm run seed
```

 Configuration

Adding/Editing Items

Edit [client/src/data/items.js](#):

```
export const items = {
  main: [
    {
      id: 1,
      name: {
        mm: 'Myanmar name',
        en: 'English name',
        zh: 'Chinese name'
      },
      image: '/webp/item-image.webp' // or emoji '🧠'
    },
    // ... more items
  ],
}
```

WhatsApp Phone Number

Edit [client/src/App.jsx](#) line 69:

```
sendQuantityWhatsApp(user.username, braisedPork, kongBak, period, '012-8533050')
```

Change '012-8533050' to your desired number.

Service Worker Version

Edit `client/public/sw.js`:

```
const CACHE_VERSION = '5.4' // Increment on each deployment
```

🔧 Deployment

See **DEPLOY.md** for complete deployment guide.

Quick Deploy

Frontend (Vercel)

```
cd client
vercel --prod --yes --build-env VITE_API_URL=https://stockcheck-
api.onrender.com/api
```

Backend (Render)

Backend auto-deploys when you push to GitHub:

```
git add .
git commit -m "Your changes"
git push
```

Render detects the push and redeploys automatically.

📄 Documentation

- [USER_GUIDE.md](#) - For end users (how to install and use)
- [OWNER_GUIDE.md](#) - For administrators (managing users, items, deployment)
- [DEPLOY.md](#) - Detailed deployment instructions
- [TODO.md](#) - Development checklist and features

🔑 API Endpoints

Authentication

- `POST /api/auth/login` - User login
- `POST /api/auth/logout` - User logout

Submissions

- `POST /api/submissions` - Create submission
- `GET /api/submissions/today` - Get all today's submissions
- `GET /api/submissions/today/:userId` - Check if user submitted today

Quantities

- `POST /api/quantities` - Save unit check quantities
- `GET /api/quantities/status` - Get current quantities status

Health

- `GET /api/health` - Backend health check
-

Technology Stack

Frontend

- **React** 18.3.1 - UI framework
- **Vite** 7.2.4 - Build tool
- **Tailwind CSS** 4.0.1 - Styling
- **FontAwesome** - Icons
- **Service Worker** - PWA functionality

Backend

- **Node.js** - Runtime
- **Express** 5.1.0 - Web framework
- **bcryptjs** - Password hashing
- **cors** - CORS middleware
- **dotenv** - Environment variables

Deployment

- **Vercel** - Frontend hosting
 - **Render** - Backend hosting
 - **GitHub** - Version control & auto-deploy
-

Multilingual Support

The app supports 3 languages:

- MM **Myanmar (မြန်မာ)** - Default for most users

- GB **English** - International users
- CN **Chinese (中文)** - WhatsApp messages, specific users

Users can toggle languages using the header button.

Security Features

- Password hashing with bcrypt (cost factor 10)
 - CORS protection for API endpoints
 - Environment variables for sensitive data
 - Input validation on all forms
 - Secure localStorage session management
 - HTTPS enforced in production (Vercel/Render)
-

PWA Features

- **Installable** - Add to home screen on any device
 - **Offline capable** - Service worker caching
 - **Auto-updates** - Version-based cache invalidation
 - **App-like experience** - Full-screen mode
 - **Fast loading** - Cached assets
 - **Responsive** - Works on all screen sizes
-

Known Limitations

Render Free Tier

- Backend restarts after 15 minutes of inactivity
- Ephemeral filesystem (quantities.json resets on restart)
- Cold start delay (~30 seconds)

Solutions:

- User/submission data persists (committed to git)
 - Quantities are meant to be temporary (morning/evening only)
 - Keep-alive not implemented (unnecessary for use case)
-

Contributing

This is a private project. If you have access and want to contribute:

1. Create a feature branch
 2. Make your changes
 3. Test thoroughly
 4. Submit pull request
-

📞 Support

For issues or questions:

1. Check documentation (USER_GUIDE.md, OWNER_GUIDE.md)
 2. Review TODO.md for known issues
 3. Contact project administrator
-

📄 License

ISC License - See LICENSE file for details

⌚ Future Enhancements

Potential improvements (see TODO.md for details):

- Firebase migration for real-time sync
 - Push notifications for submissions
 - Export submissions to Excel/CSV
 - Advanced analytics dashboard
 - Role-based permissions (admin/user)
-

📊 Project Status

- **Phase 1-14:** Completed
 - **Phase 15:** Documentation complete
 - **Status:** Production ready
 - **Version:** 5.4
 - **Live:** <https://client-scydom-chins-projects.vercel.app>
-

Built with ❤️ for efficient stock management

Last Updated: December 2025