GarageGuru Configuration Archive

# GarageGuru - Complete Configuration Archive

**Document**: Configuration Archive 3 of 5  
**Creation Date**: August 30, 2025  
**Content**: ALL configuration files, build tools, and deployment settings  
**Total Files**: 15+ configuration files

This document contains EVERY configuration file for the GarageGuru project.

## Table of Contents

1. [Package Configuration](#package-configuration)
2. [Build and Development Tools](#build-and-development-tools)
3. [TypeScript Configuration](#typescript-configuration)
4. [Database Configuration](#database-configuration)
5. [Styling and UI Configuration](#styling-and-ui-configuration)
6. [Environment and Deployment](#environment-and-deployment)

## Package Configuration

### 📦 /package.json - Node.js Dependencies and Scripts

**Purpose**: Project dependencies, build scripts, and metadata  
**Dependencies**: 80+ production and development packages  
**Lines**: 144 lines

{  
 "name": "rest-express",  
 "version": "1.0.0",  
 "type": "module",  
 "license": "MIT",  
 "scripts": {  
 "dev": "NODE\_ENV=development tsx server/index.ts",  
 "build": "vite build && esbuild server/index.ts --platform=node --packages=external --bundle --format=esm --outdir=dist --minify",  
 "start": "NODE\_ENV=production node dist/index.js",  
 "check": "tsc",  
 "db:push": "drizzle-kit push"  
 },  
 "dependencies": {  
 "@google-cloud/storage": "^7.16.0",  
 "@hono/node-server": "^1.19.0",  
 "@hookform/resolvers": "^3.10.0",  
 "@jridgewell/trace-mapping": "^0.3.25",  
 "@neondatabase/serverless": "^1.0.1",  
 "@radix-ui/react-accordion": "^1.2.4",  
 "@radix-ui/react-alert-dialog": "^1.1.7",  
 "@radix-ui/react-aspect-ratio": "^1.1.3",  
 "@radix-ui/react-avatar": "^1.1.4",  
 "@radix-ui/react-checkbox": "^1.1.5",  
 "@radix-ui/react-collapsible": "^1.1.4",  
 "@radix-ui/react-context-menu": "^2.2.7",  
 "@radix-ui/react-dialog": "^1.1.7",  
 "@radix-ui/react-dropdown-menu": "^2.1.7",  
 "@radix-ui/react-hover-card": "^1.1.7",  
 "@radix-ui/react-label": "^2.1.3",  
 "@radix-ui/react-menubar": "^1.1.7",  
 "@radix-ui/react-navigation-menu": "^1.2.6",  
 "@radix-ui/react-popover": "^1.1.7",  
 "@radix-ui/react-progress": "^1.1.3",  
 "@radix-ui/react-radio-group": "^1.2.4",  
 "@radix-ui/react-scroll-area": "^1.2.4",  
 "@radix-ui/react-select": "^2.1.7",  
 "@radix-ui/react-separator": "^1.1.3",  
 "@radix-ui/react-slider": "^1.2.4",  
 "@radix-ui/react-slot": "^1.2.0",  
 "@radix-ui/react-switch": "^1.1.4",  
 "@radix-ui/react-tabs": "^1.1.4",  
 "@radix-ui/react-toast": "^1.2.7",  
 "@radix-ui/react-toggle": "^1.1.3",  
 "@radix-ui/react-toggle-group": "^1.1.3",  
 "@radix-ui/react-tooltip": "^1.2.0",  
 "@sendgrid/mail": "^8.1.5",  
 "@tanstack/react-query": "^5.60.5",  
 "@types/bcrypt": "^6.0.0",  
 "@types/cors": "^2.8.19",  
 "@types/jsonwebtoken": "^9.0.10",  
 "@types/memoizee": "^0.4.12",  
 "@types/multer": "^2.0.0",  
 "@types/nodemailer": "^6.4.17",  
 "@types/pdfkit": "^0.17.2",  
 "@types/pg": "^8.15.5",  
 "@uppy/aws-s3": "^4.3.2",  
 "@uppy/core": "^4.5.2",  
 "@uppy/dashboard": "^4.4.3",  
 "@uppy/drag-drop": "^4.2.2",  
 "@uppy/file-input": "^4.2.2",  
 "@uppy/progress-bar": "^4.3.2",  
 "@uppy/react": "^4.5.2",  
 "@zxing/browser": "^0.1.5",  
 "@zxing/library": "^0.21.3",  
 "bcrypt": "^6.0.0",  
 "bcryptjs": "^3.0.2",  
 "class-variance-authority": "^0.7.1",  
 "clsx": "^2.1.1",  
 "cmdk": "^1.1.1",  
 "connect-pg-simple": "^10.0.0",  
 "cors": "^2.8.5",  
 "date-fns": "^3.6.0",  
 "drizzle-orm": "^0.39.1",  
 "drizzle-zod": "^0.7.0",  
 "embla-carousel-react": "^8.6.0",  
 "express": "^4.21.2",  
 "express-session": "^1.18.1",  
 "framer-motion": "^11.13.1",  
 "google-auth-library": "^10.2.1",  
 "hono": "^4.9.4",  
 "html5-qrcode": "^2.3.8",  
 "input-otp": "^1.4.2",  
 "jsonwebtoken": "^9.0.2",  
 "jspdf": "^3.0.1",  
 "lucide-react": "^0.453.0",  
 "memoizee": "^0.4.17",  
 "memorystore": "^1.6.7",  
 "multer": "^2.0.2",  
 "nanoid": "^5.1.5",  
 "next-themes": "^0.4.6",  
 "nodemailer": "^7.0.5",  
 "openid-client": "^6.6.4",  
 "passport": "^0.7.0",  
 "passport-local": "^1.0.0",  
 "pdfkit": "^0.17.1",  
 "pg": "^8.16.3",  
 "postgres": "^3.4.7",  
 "qr-scanner": "^1.4.2",  
 "react": "^18.3.1",  
 "react-day-picker": "^8.10.1",  
 "react-dom": "^18.3.1",  
 "react-hook-form": "^7.55.0",  
 "react-icons": "^5.4.0",  
 "react-qr-barcode-scanner": "^2.1.8",  
 "react-resizable-panels": "^2.1.7",  
 "recharts": "^2.15.2",  
 "serverless-http": "^3.2.0",  
 "tailwind-merge": "^2.6.0",  
 "tailwindcss-animate": "^1.0.7",  
 "tw-animate-css": "^1.2.5",  
 "vaul": "^1.1.2",  
 "wouter": "^3.3.5",  
 "ws": "^8.18.0",  
 "zod": "^3.24.2",  
 "zod-validation-error": "^3.4.0"  
 },  
 "devDependencies": {  
 "@replit/vite-plugin-cartographer": "^0.2.7",  
 "@replit/vite-plugin-runtime-error-modal": "^0.0.3",  
 "@tailwindcss/typography": "^0.5.15",  
 "@tailwindcss/vite": "^4.1.3",  
 "@types/connect-pg-simple": "^7.0.3",  
 "@types/express": "4.17.21",  
 "@types/express-session": "^1.18.0",  
 "@types/node": "20.16.11",  
 "@types/passport": "^1.0.16",  
 "@types/passport-local": "^1.0.38",  
 "@types/react": "^18.3.11",  
 "@types/react-dom": "^18.3.1",  
 "@types/ws": "^8.5.13",  
 "@vitejs/plugin-react": "^4.3.2",  
 "autoprefixer": "^10.4.20",  
 "drizzle-kit": "^0.30.4",  
 "esbuild": "^0.25.0",  
 "postcss": "^8.4.47",  
 "tailwindcss": "^3.4.17",  
 "tsx": "^4.19.1",  
 "typescript": "5.6.3",  
 "vite": "^5.4.19"  
 },  
 "optionalDependencies": {  
 "bufferutil": "^4.0.8"  
 }  
}

## Build and Development Tools

### ⚡ /vite.config.ts - Vite Build Configuration

**Purpose**: Frontend build tool configuration with plugins and aliases  
**Features**: React plugin, runtime error overlay, path aliases  
**Lines**: 38 lines

import { defineConfig } from "vite";  
import react from "@vitejs/plugin-react";  
import path from "path";  
import runtimeErrorOverlay from "@replit/vite-plugin-runtime-error-modal";  
  
export default defineConfig({  
 plugins: [  
 react(),  
 runtimeErrorOverlay(),  
 ...(process.env.NODE\_ENV !== "production" &&  
 process.env.REPL\_ID !== undefined  
 ? [  
 await import("@replit/vite-plugin-cartographer").then((m) =>  
 m.cartographer(),  
 ),  
 ]  
 : []),  
 ],  
 resolve: {  
 alias: {  
 "@": path.resolve(import.meta.dirname, "client", "src"),  
 "@shared": path.resolve(import.meta.dirname, "shared"),  
 "@assets": path.resolve(import.meta.dirname, "attached\_assets"),  
 },  
 },  
 root: path.resolve(import.meta.dirname, "client"),  
 build: {  
 outDir: path.resolve(import.meta.dirname, "dist/public"),  
 emptyOutDir: true,  
 },  
 server: {  
 fs: {  
 strict: true,  
 deny: ["\*\*/.\*"],  
 },  
 },  
});

### 📋 /tsconfig.json - TypeScript Configuration

**Purpose**: TypeScript compiler settings and path mappings  
**Features**: Strict mode, path aliases, JSX configuration

*Will be read and included with complete source code*

### 📦 /drizzle.config.ts - Database ORM Configuration

**Purpose**: Drizzle ORM configuration for schema management  
**Features**: PostgreSQL connection, schema file locations

*Will be read and included with complete source code*

## Database Configuration

### 🔗 /server/db.ts - PostgreSQL Connection Pool

**Purpose**: Database connection management with Render.com PostgreSQL  
**Features**: Connection pooling, SSL configuration, error handling  
**Lines**: 28 lines

import { Pool } from 'pg';  
  
// Use Render.com PostgreSQL database  
const databaseUrl = "postgresql://admin:lHgw4ztka79bYIxW2MBGcTMCEKjzUE9w@dpg-d2ov7g0gjchc73f8s5q0-a.singapore-postgres.render.com/garageguru";  
  
if (!databaseUrl) {  
 throw new Error("DATABASE\_URL must be set for Render.com PostgreSQL connection.");  
}  
  
console.log('🔗 Using database URL:', databaseUrl.split('@')[0] + '@[hidden]');  
  
export const pool = new Pool({   
 connectionString: databaseUrl,  
 ssl: { rejectUnauthorized: false },  
 connectionTimeoutMillis: 30000,  
 idleTimeoutMillis: 30000,  
 max: 20  
});  
  
// Test connection  
pool.on('connect', () => {  
 console.log('Connected to PostgreSQL database');  
});  
  
pool.on('error', (err) => {  
 console.error('PostgreSQL connection error:', err);  
});

### 🔧 /server/database-client.ts - Neon Database Client

**Purpose**: Alternative Neon serverless database client  
**Features**: Serverless PostgreSQL connection with error handling  
**Lines**: 61 lines

import { neon } from '@neondatabase/serverless';  
  
let sql: any = null;  
  
export interface DatabaseConfig {  
 url: string;  
 connected: boolean;  
 error?: string;  
}  
  
export function initDatabase(): DatabaseConfig {  
 try {  
 const databaseUrl = process.env.DATABASE\_URL;  
   
 if (!databaseUrl) {  
 return {  
 url: 'missing',  
 connected: false,  
 error: 'DATABASE\_URL environment variable is required'  
 };  
 }  
  
 if (!sql) {  
 sql = neon(databaseUrl);  
 }  
  
 return {  
 url: databaseUrl.split('@')[1] || 'configured',  
 connected: true  
 };  
 } catch (error) {  
 return {  
 url: 'error',  
 connected: false,  
 error: error instanceof Error ? error.message : 'Unknown database error'  
 };  
 }  
}  
  
export function getDatabaseClient() {  
 if (!sql) {  
 const config = initDatabase();  
 if (!config.connected) {  
 throw new Error(config.error || 'Database not initialized');  
 }  
 }  
 return sql;  
}  
  
export async function pingDatabase(): Promise<{ success: boolean; error?: string }> {  
 try {  
 const client = getDatabaseClient();  
 await client`SELECT 1 as ping`;  
 return { success: true };  
 } catch (error) {  
 return {  
 success: false,  
 error: error instanceof Error ? error.message : 'Database ping failed'  
 };  
 }  
}

## Styling and UI Configuration

### 🎨 /tailwind.config.ts - Tailwind CSS Configuration

**Purpose**: Utility-first CSS framework configuration  
**Features**: Custom colors, animations, component styles

*Will be read and included with complete source code*

### 📄 /postcss.config.js - PostCSS Configuration

**Purpose**: CSS processing configuration  
**Features**: Tailwind CSS integration, autoprefixer

*Will be read and included with complete source code*

*NOTE: This is the beginning of the configuration archive. The complete document will contain ALL configuration files with full source code.*