ตัวอย่าง Environment Variables สำหรับ Admin Setup

📁 ไฟล์ .env

```
# ==============================================
# DATABASE CONFIGURATION
# ==============================================
DATABASE_URL="postgresql://username:password@localhost:5432/
sss_surplus_db"
# หรือ
DATABASE_URL="mysql://username:password@localhost:3306/
sss_surplus_db"

# ==============================================
# ADMIN CREDENTIALS
# ==============================================

# Super Admin Account (ระดับสูงสุด)
SUPER_ADMIN_EMAIL="sanchai565@gmail.com"
SUPER_ADMIN_PASSWORD="SuperAdmin@2024!"
SUPER_ADMIN_NAME="Super Administrator"

# Default Admin Account (Admin ทั่วไป)
ADMIN_EMAIL="admin@sss-surplus.com"
ADMIN_PASSWORD="AdminSSS@2024!"
ADMIN_NAME="System Administrator"

# Alternative Admin Accounts
ADMIN_EMAIL_2="manager@sss-surplus.com"
ADMIN_PASSWORD_2="Manager@2024!"

# ==============================================
# AUTHENTICATION SETTINGS
# ==============================================

# JWT Secret สำหรับ Admin
ADMIN_JWT_SECRET="your-super-secret-admin-jwt-key-here-2024"
ADMIN_JWT_EXPIRES_IN="24h"

# Session Secret
ADMIN_SESSION_SECRET="your-admin-session-secret-key-here"

# Password Hash Salt Rounds
```

```
BCRYPT_SALT_ROUNDS=12

# ==============================================
# ADMIN PANEL SETTINGS
# ==============================================

# Admin Panel URL Path
ADMIN_PATH="/admin"
SUPER_ADMIN_PATH="/admin/super"

# Admin Panel Security
ADMIN_2FA_ENABLED=true
ADMIN_LOGIN_ATTEMPTS_LIMIT=5
ADMIN_LOGIN_LOCKOUT_DURATION=30

# ==============================================
# EMAIL CONFIGURATION (สำหรับ Admin Notifications)
# ==============================================
ADMIN_EMAIL_HOST="smtp.gmail.com"
ADMIN_EMAIL_PORT=587
ADMIN_EMAIL_USER="admin-notifications@sss-surplus.com"
ADMIN_EMAIL_PASS="your-email-app-password"
ADMIN_EMAIL_FROM="SSS Surplus Admin <admin@sss-surplus.com>"

# ==============================================
# SECURITY SETTINGS
# ==============================================

# Admin IP Whitelist (ถ้าต้องการจำกัด IP)
ADMIN_ALLOWED_IPS="192.168.1.100,203.154.xxx.xxx"

# Admin Access Hours (24h format)
ADMIN_ACCESS_START_HOUR=8
ADMIN_ACCESS_END_HOUR=22

# ==============================================
# VENDOR & CUSTOMER SETTINGS
# ==============================================

# Default Vendor Commission Rate
DEFAULT_VENDOR_COMMISSION=3.0

# Customer Registration Settings
CUSTOMER_EMAIL_VERIFICATION=true
VENDOR_MANUAL_APPROVAL=true

# ==============================================
# PAYMENT GATEWAY
# ==============================================
PAYMENT_GATEWAY_API_KEY="your-payment-gateway-api-key"
PAYMENT_GATEWAY_SECRET="your-payment-gateway-secret"
```

```
# ==============================================
# FILE UPLOAD SETTINGS
# ==============================================
MAX_FILE_SIZE=10485760  # 10MB
ALLOWED_FILE_TYPES="jpg,jpeg,png,pdf,doc,docx"
UPLOAD_PATH="/uploads"

# ==============================================
# NOTIFICATION SETTINGS
# ==============================================

# LINE Notify (สำหรับแจ้งเตือน Admin)
LINE_NOTIFY_TOKEN="your-line-notify-token"

# SMS Gateway
SMS_API_KEY="your-sms-api-key"
SMS_SENDER="SSS-Surplus"

# ==============================================
# DEVELOPMENT/PRODUCTION FLAGS
# ==============================================
NODE_ENV="production"
DEBUG_MODE=false
ADMIN_DEBUG=false

# ==============================================
# BACKUP SETTINGS
# ==============================================
AUTO_BACKUP_ENABLED=true
BACKUP_SCHEDULE="0 2 * * *"  # ทุกวัน 2:00 AM
BACKUP_RETENTION_DAYS=30
```

🔧 ตัวอย่างการใช้งานใน Code

1. Next.js - lib/auth.js

```
 // lib/auth.js
export const adminCredentials = {
  superAdmin: {
    email: process.env.SUPER_ADMIN_EMAIL,
    password: process.env.SUPER_ADMIN_PASSWORD,
    name: process.env.SUPER_ADMIN_NAME,
    role: 'super_admin'
  },
  admin: {
    email: process.env.ADMIN_EMAIL,
    password: process.env.ADMIN_PASSWORD,
    name: process.env.ADMIN_NAME,
    role: 'admin'
  }
};

export const authConfig = {
  jwtSecret: process.env.ADMIN_JWT_SECRET,
  jwtExpiresIn: process.env.ADMIN_JWT_EXPIRES_IN,
  sessionSecret: process.env.ADMIN_SESSION_SECRET,
  saltRounds: parseInt(process.env.BCRYPT_SALT_ROUNDS) || 12
};
```

2. Database Seeder - prisma/seed.js

```
// prisma/seed.js
import bcrypt from 'bcryptjs';
import { PrismaClient } from '@prisma/client';

const prisma = new PrismaClient();

async function seedAdmins() {
  // Super Admin
  const superAdminPassword = await bcrypt.hash(
    process.env.SUPER_ADMIN_PASSWORD,
    parseInt(process.env.BCRYPT_SALT_ROUNDS)
  );

  await prisma.admin.upsert({
    where: { email: process.env.SUPER_ADMIN_EMAIL },
    update: {},
    create: {
      email: process.env.SUPER_ADMIN_EMAIL,
      password: superAdminPassword,
      name: process.env.SUPER_ADMIN_NAME,
      role: 'SUPER_ADMIN',
      isActive: true,
      createdAt: new Date()
    }
  });

  // Regular Admin
  const adminPassword = await bcrypt.hash(
    process.env.ADMIN_PASSWORD,
    parseInt(process.env.BCRYPT_SALT_ROUNDS)
  );

  await prisma.admin.upsert({
    where: { email: process.env.ADMIN_EMAIL },
    update: {},
    create: {
      email: process.env.ADMIN_EMAIL,
      password: adminPassword,
      name: process.env.ADMIN_NAME,
      role: 'ADMIN',
      isActive: true,
      createdAt: new Date()
    }
  });
}

seedAdmins()
  .catch((e) => {
    console.error(e);
    process.exit(1);
  })
  .finally(async () => {
    await prisma.$disconnect();
  });
```

3. API Route - pages/api/admin/login.js

```
// pages/api/admin/login.js
import bcrypt from 'bcryptjs';
import jwt from 'jsonwebtoken';
import { adminCredentials, authConfig } from '../../../lib/
auth';

export default async function handler(req, res) {
  if (req.method !== 'POST') {
    return res.status(405).json({ message: 'Method not
allowed' });
  }

  const { email, password } = req.body;

  try {
    // ตรวจสอบ Super Admin
    if (email === adminCredentials.superAdmin.email) {
      const isValid = await bcrypt.compare(password,
adminCredentials.superAdmin.password);
      if (isValid) {
        const token = jwt.sign(
          {
            email: email,
            role: 'super_admin',
            name: adminCredentials.superAdmin.name
          },
          authConfig.jwtSecret,
          { expiresIn: authConfig.jwtExpiresIn }
        );
        return res.status(200).json({
          success: true,
          token,
          user: {
            email: email,
            name: adminCredentials.superAdmin.name,
            role: 'super_admin'
          },
          redirectTo: '/admin/super/dashboard'
        });
      }
    }

    // ตรวจสอบ Admin ทั่วไป
    if (email === adminCredentials.admin.email) {
      const isValid = await bcrypt.compare(password,
adminCredentials.admin.password);
      if (isValid) {
        const token = jwt.sign(
          {
            email: email,
            role: 'admin',
            name: adminCredentials.admin.name
          },
          authConfig.jwtSecret,
          { expiresIn: authConfig.jwtExpiresIn }
        );

        return res.status(200).json({
          success: true,
          token,
          user: {
            email: email,
            name: adminCredentials.admin.name,
            role: 'admin'
          },
          redirectTo: '/admin/dashboard'
        });
      }
    }

    return res.status(401).json({
      success: false,
      message: 'Invalid credentials'
    });

  } catch (error) {
    console.error('Login error:', error);
    return res.status(500).json({
      success: false,
      message: 'Internal server error'
    });
  }
}
```

🚀 คำสั่งสำหรับ Setup

1. สร้าง Admin ครั้งแรก

```bash
# ถ้าใช้ Prisma
npx prisma db seed

# ถ้าใช้ custom script
npm run seed:admin

# ถ้าใช้ Next.js
npm run setup:admin
```

2. Hash Password ใหม่

```bash
# สร้าง script สำหรับ hash password
node scripts/hash-password.js "YourNewPassword123!"
```

3. ตรวจสอบ Admin ใน Database

```sql
-- ดู Admin ทั้งหมด
SELECT id, email, name, role, is_active, created_at
FROM admins;

-- อัปเดต Password
UPDATE admins
SET password = '$2b$12$hashedPasswordHere'
WHERE email = 'admin@sss-surplus.com';
```

⚠️ หมายเหตุความปลอดภัย

1. **เปลี่ยน Password**: เปลี่ยน default passwords ทันทีหลัง setup
2. **Strong Passwords**: ใช้ password ที่มีความซับซ้อนสูง
3. **Environment Security**: ไม่ commit ไฟล์ .env เข้า Git
4. **Regular Updates**: อัปเดต credentials เป็นระยะ
5. **Access Logs**: เปิดใช้ logging สำหรับ admin access