# **CyberSentinel Email Security Dashboard**

## **Deployment Manual**

## **Table of Contents**

1. [System Requirements](https://claude.ai/chat/9961578f-75f5-47b4-bf09-df3e2847bbf7#system-requirements)
2. [Installation](https://claude.ai/chat/9961578f-75f5-47b4-bf09-df3e2847bbf7#installation)
3. [Configuration](https://claude.ai/chat/9961578f-75f5-47b4-bf09-df3e2847bbf7#configuration)
4. [Database Setup](https://claude.ai/chat/9961578f-75f5-47b4-bf09-df3e2847bbf7#database-setup)
5. [Security Configuration](https://claude.ai/chat/9961578f-75f5-47b4-bf09-df3e2847bbf7#security-configuration)
6. [Testing](https://claude.ai/chat/9961578f-75f5-47b4-bf09-df3e2847bbf7#testing)
7. [Deployment](https://claude.ai/chat/9961578f-75f5-47b4-bf09-df3e2847bbf7#deployment)
8. [Updating](https://claude.ai/chat/9961578f-75f5-47b4-bf09-df3e2847bbf7#updating)

## **System Requirements**

### **Minimum Requirements**

* Node.js v16.0 or higher
* npm v8.0 or higher
* Next.js v14.0 or higher
* 2GB RAM (4GB recommended)
* 10GB storage space
* Modern web browser (Chrome, Firefox, Edge, Safari)

### **Recommended Development Environment**

* Visual Studio Code with ESLint and Prettier
* Git for version control
* Node.js v18.0 or higher
* npm v9.0 or higher

## **Installation**

### **Local Development Setup**

* Clone the repository:  
    
   git clone https://github.com/htmw/2025S-Tech-Titans
* cd 2025S-Tech-Titans/cybersentinel
* Install dependencies:  
    
   npm install

1. Set up environment variables:  
   * Create a .env.local file in the project root directory
   * Add the required environment variables (see Configuration section)

* Start the development server:  
    
   npm run dev

1. Access the application at http://localhost:3000

## **Configuration**

### **Environment Variables**

Create a .env.local file with the following variables:

* # ML API Key for Email Analysis
* ML\_API\_KEY=your\_ml\_api\_key
* # Next.js Configuration
* NEXT\_PUBLIC\_APP\_URL=http://localhost:3000
* # Local Storage Prefix (for multi-tenant deployment)
* NEXT\_PUBLIC\_STORAGE\_PREFIX=cybersentinel\_
* # Optional: Analytics Integration
* NEXT\_PUBLIC\_ANALYTICS\_ID=your\_analytics\_id

### **Project Structure**

Key directories and files:

* /app: Next.js app router setup
* /components: React components
* /lib: Utility functions and contexts
* /public: Static assets

## **Database Setup**

CyberSentinel uses localStorage for data persistence in the default configuration. For production deployments, you should implement a proper database solution.

### **Setting up MongoDB (Recommended)**

1. Create a MongoDB Atlas account or set up a local MongoDB server
2. Create a new database named cybersentinel

* Add the MongoDB connection string to your .env.local file:  
    
   MONGODB\_URI=your\_mongodb\_connection\_string

1. Update the storage providers in /lib directory to use MongoDB instead of localStorage:  
   * Modify auth-context.tsx
   * Modify global-storage.ts
   * Modify notifications-store.ts

## **Security Configuration**

### **API Key Security**

1. Store your ML API key securely in environment variables
2. Never expose your API key in client-side code
3. Set up proper request validation in your API routes

### **Authentication**

The default implementation uses a simple localStorage-based authentication system. For production:

* Implement a more secure authentication provider like NextAuth.js:  
    
   npm install next-auth

1. Configure authentication providers in /app/api/auth/[...nextauth]/route.ts
2. Update the AuthProvider component to use NextAuth

### **HTTPS Setup**

For production deployment, always use HTTPS:

1. Obtain an SSL certificate (e.g., from Let's Encrypt)
2. Configure your hosting provider to use HTTPS
3. Update the NEXT\_PUBLIC\_APP\_URL environment variable to use https://

## **Testing**

### **Running Tests**

Run the test suite to ensure everything is working:

* npm test

### **Manual Testing Checklist**

* Verify user registration and login
* Test email analysis functionality with the ML API
* Check all security tools
* Verify permissions for each role
* Test notifications system
* Ensure responsive design works on various devices

## **Deployment**

### **Deploying to Vercel (Recommended)**

1. Create a Vercel account and connect it to your GitHub repository
2. Add all environment variables in the Vercel project settings

* Deploy using the Vercel dashboard or CLI:  
   npm install -g vercelvercel

### **Deploying to Other Platforms**

For other platforms (AWS, Azure, DigitalOcean, etc.):

* Build the production version:  
    
   npm run build
* Test the production build locally:  
    
   npm start

1. Deploy the .next directory and necessary server files according to your hosting provider's instructions

## **Updating**

### **Updating Dependencies**

Regularly update dependencies to ensure security and performance:

* npm update

For major updates, follow these steps:

1. Check the changelog for breaking changes
2. Update dependencies one at a time
3. Run tests after each update
4. Fix any issues before proceeding to the next dependency

### **Version Control**

Follow these best practices for version management:

1. Use semantic versioning (MAJOR.MINOR.PATCH)
2. Create release tags for each version
3. Maintain a changelog documenting all significant changes