WebShield AI - Project Specification Document

□ Project Overview

Project Name: WebShield AI

Version: 1.0.0

Status: Production Ready

Last Updated: December 2024

Project Type: SaaS Web Application

License: Free Forever (Community Supported)

□ Project Mission

WebShield AI is a comprehensive, free security and compliance scanning platform designed to democratize cybersecurity tools. Our mission is to make enterprise-grade security scanning accessible to everyone, regardless of budget, while maintaining the highest standards of accuracy and reliability.

□ Core Features

1. AI-Powered Compliance Scanner

- **WCAG 2.1 Compliance** - Automated accessibility testing

- **GDPR & Privacy - Data protection compliance scanning**

- **HIPAA Standards - Healthcare compliance verification**

- **Real-time Analysis - Instant compliance scoring**

- **99.9% Accuracy Rate - Industry-leading precision**

2. Security Engine Suite (12+ Tools)

Active Tools (Real-time, No API Keys Required):

- **Port Scanner - TCP/UDP port scanning with service detection**

- **SSL Certificate Checker - SSL/TLS certificate validation and analysis**

- **DNS Analyzer - Comprehensive DNS security assessment**

- **Password Strength Analyzer - Entropy-based password security analysis**

- **Hash Identifier & Analyzer - Hash type detection and security assessment**

Coming Soon Tools:

- HTTP Header Analyzer

- Subdomain Finder

- WHOIS Lookup

- Robots.txt Analyzer

- Email Validator

- IP Geolocation

- Hash Analyzer

- URL Analyzer

- File Analyzer

3. Project Management

- **Unlimited Projects - Create and manage multiple websites**

- **Team Collaboration - Role-based access controls**

- **Real-time Monitoring - Continuous website monitoring**

- **Scheduled Scans - Automated compliance checking**

- **Comprehensive Reports - Detailed analysis and recommendations**

4. Advanced Analytics

- **Compliance Scores** - Real-time scoring system

- **Security Metrics** - Vulnerability assessment

- **Trend Analysis** - Historical data tracking

- **Progress Tracking** - Improvement monitoring

□ Technical Architecture

Frontend Stack

- **Framework: Next.js 15.4.6 (React 18.3.1)**

- **Language: TypeScript 5.x**

- **Styling: Tailwind CSS 3.4.17**

- **Animations: Framer Motion 12.23.12**

- **Authentication: Clerk.js 6.31.3**

- **Icons: Lucide React 0.540.0**

- **State Management: React Hooks + Context**

Backend Stack

- **Runtime: Node.js with Express.js 4.18.2**

- **Language: TypeScript 5.3.3**

- **Database: MongoDB 8.0.3 with Mongoose**

- **Authentication: JWT with Clerk integration**

- **Scheduling: Node-cron 4.2.1**

- **HTTP Client: Axios 1.11.0**

- **Compression: Express compression middleware**

Infrastructure

- **Hosting: Vercel (Frontend) / Railway/Heroku (Backend)**

- **Database: MongoDB Atlas**

- **Authentication: Clerk.com**

- **Payments: Ko-fi (Donations)**

- **Monitoring: Built-in health checks**

- **Security: HTTPS, Security Headers, CORS**

□ Security Features

Frontend Security

- **Content Security Policy - XSS protection**

- **X-Frame-Options - Clickjacking prevention**

- **X-Content-Type-Options - MIME type sniffing protection**

- **Referrer Policy - Privacy protection**

- **Permissions Policy - Feature access control**

- **Strict Transport Security - HTTPS enforcement**

Backend Security

- **JWT Authentication - Secure token-based auth**

- **Rate Limiting - API abuse prevention**

- **Input Validation - Request sanitization**

- **CORS Configuration - Cross-origin protection**

- **Request Size Limits - DoS attack prevention**

- **Error Handling - Secure error responses**

Data Security

- **Encryption at Rest** - Database encryption

- **Encryption in Transit** - HTTPS/TLS

- **Secure Headers** - Comprehensive security headers

- **Input Sanitization** - XSS/SQL injection prevention

- **Session Management** - Secure session handling

□ Performance Optimizations

Frontend Performance

- **Code Splitting - Dynamic imports and lazy loading**

- **Bundle Optimization - Webpack optimization**

- **Image Optimization - WebP/AVIF support**

- **CSS Optimization - Purged unused styles**

- **Caching Strategy - Browser and CDN caching**

- **Lazy Loading - Component and image lazy loading**

Backend Performance

- **Compression - Gzip/Brotli compression**

- **Database Indexing - Optimized queries**

- **Connection Pooling - Efficient database connections**

- **Caching - Redis integration ready**

- **Load Balancing - Horizontal scaling support**

□ Development & Deployment

Development Environment

- **Node.js: 18.x or higher**

- **npm/yarn: Package management**

- **Git: Version control**

- **ESLint:** Code quality

- **TypeScript:** Type safety

- **Hot Reload:** Development server

Build Process

```bash

# Frontend

npm run build    # Production build

npm run dev      # Development server

npm run lint     # Code linting


# Backend

npm run build    # TypeScript compilation
```

npm run dev   # Development server

npm run start   # Production server

```

Environment Variables

```env

 # Frontend (.env.local)

NEXT_PUBLIC_CLERK_PUBLISHABLE_KEY=your_clerk_key

NEXT_PUBLIC_API_URL=http://localhost:3001

 # Backend (.env)

MONGODB_URI=your_mongodb_uri

CLERK_SECRET_KEY=your_clerk_secret

NODE_ENV=production

PORT=3001

```

Scalability Features

Horizontal Scaling

- **Stateless Backend - Session-less architecture**

- **Database Sharding - MongoDB sharding support**

- **Load Balancing - Multiple server instances**

- **CDN Integration - Global content delivery**

- **Microservices Ready - Modular architecture**

Performance Monitoring

- **Health Checks - `/health` endpoint**

- **Error Tracking - Comprehensive error logging**

- **Performance Metrics - Response time monitoring**

- **Resource Usage - Memory and CPU tracking**

- **Uptime Monitoring - Service availability**

□ User Interface

Design System

- **Color Palette:** Blue gradient theme with accessibility compliance

- **Typography:** Inter font family

- **Components:** Reusable UI components

- **Responsive Design:** Mobile-first approach

- **Dark Mode:** System preference support

- **Accessibility:** WCAG 2.1 AA compliance

Key Pages

1. **Landing Page** - Feature showcase and pricing

2. **Dashboard** - Project overview and analytics

3. **Security Engine** - Security tools interface

4. **Projects** - Website management

5. **Reports** - Detailed analysis reports

6. **Settings** - User preferences and account

7. **About** - Company information

8. **Donation** - Ko-fi integration

□ API Endpoints

Authentication

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

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

- `GET /api/auth/verify` - Token verification

Projects

- `GET /api/projects` - List user projects

- `POST /api/projects` - Create new project

- `PUT /api/projects/:id` - Update project

- `DELETE /api/projects/:id` - Delete project

Scans

- `GET /api/scans` - List scans

- `POST /api/scans` - Start new scan

- `GET /api/scans/:id` - Get scan results

- `DELETE /api/scans/:id` - Delete scan

Monitoring

- `GET /api/monitoring` - List monitored websites

- `POST /api/monitoring` - Add website monitoring

- `PUT /api/monitoring/:id` - Update monitoring

- `DELETE /api/monitoring/:id` - Remove monitoring

□ Database Schema

User Model

```typescript
interface User {
  _id: ObjectId;
  clerkId: string;
  email: string;
  name: string;
  isSupporter: boolean;
  createdAt: Date;
  updatedAt: Date;
```

}

```

Project Model

```typescript

interface Project {

_id: ObjectId;

userId: ObjectId;

name: string;

url: string;

description?: string;

isActive: boolean;

createdAt: Date;

```
  updatedAt: Date;
}
```

Scan Model

```typescript
interface Scan {
  _id: ObjectId;
  projectId: ObjectId;
  userId: ObjectId;
  status: 'pending' | 'scanning' | 'completed' | 'failed';
  results: ScanResults;
```

```
  startedAt: Date;
  completedAt?: Date;
  createdAt: Date;
}
```

□ Deployment Guide

Frontend Deployment (Vercel)

1. Connect GitHub repository to Vercel
2. Configure environment variables
3. Set build command: `npm run build`
4. Deploy automatically on push

Backend Deployment (Railway/Heroku)

1. Connect GitHub repository

2. Set environment variables

3. Configure build command: `npm run build && npm start`

4. Set Node.js version: 18.x

Database Setup (MongoDB Atlas)

1. Create MongoDB Atlas cluster

2. Configure network access

3. Create database user

4. Get connection string

5. Set environment variable

□ Testing Strategy

Frontend Testing

- **Unit Tests: Jest + React Testing Library**

- **E2E Tests: Playwright**

- **Component Tests: Storybook ready**

- **Accessibility Tests: axe-core integration**

Backend Testing

- **Unit Tests: Jest**

- **Integration Tests: Supertest**

- **API Tests: Postman collections**

- **Load Testing: Artillery ready**

□ Analytics & Monitoring

User Analytics

- **Page Views - User journey tracking**

- **Feature Usage - Tool adoption metrics**

- **Error Tracking - Bug monitoring**

- **Performance - Core Web Vitals**

Business Metrics

- **User Growth** - Registration trends

- **Retention** - User engagement

- **Conversion** - Donation rates

- **Support** - Community feedback

□ Future Roadmap

Phase 1 (Q1 2025)

- [ ] Additional security tools

- [ ] Advanced reporting features

- [ ] Team collaboration enhancements

- [ ] API rate limiting improvements

Phase 2 (Q2 2025)

- [ ] Mobile application
- [ ] Advanced AI features
- [ ] Enterprise integrations
- [ ] White-label solutions

Phase 3 (Q3 2025)

- [ ] Advanced compliance frameworks
- [ ] Real-time collaboration
- [ ] Advanced analytics dashboard
- [ ] Custom rule engine

□ Contributing

Development Setup

1. Fork the repository

2. Create feature branch

3. Make changes with tests

4. Submit pull request

5. Code review process

Code Standards

- **TypeScript** - Strict type checking

- **ESLint** - Code quality rules

- **Prettier** - Code formatting

- **Conventional Commits - Git commit messages**

□ Support & Contact

Technical Support

- **Documentation: Comprehensive guides**

- **GitHub Issues: Bug reports and features**

- **Community: Discord server**

- **Email: support@webshield.ai**

Business Inquiries

- **Partnerships: partnerships@webshield.ai**

- **Enterprise: enterprise@webshield.ai**

- **Media: press@webshield.ai**

□ License & Legal

License

- **Free Forever - No cost for core features**

- **Open Source - Community contributions welcome**

- **MIT License - Permissive licensing**

Privacy

- **GDPR Compliant** - Data protection

- **Privacy Policy** - User data handling

- **Terms of Service** - Usage guidelines

- **Cookie Policy** - Tracking transparency

Document Version: 1.0.0

Last Updated: December 2024

Maintained By: WebShield AI Team

Contact: support@webshield.ai