Web Interface Setup Guide

Overview

The SSVproff Web Interface is a Next.js-based frontend application that connects to the FastAPI backend. This guide will help you set up and run the web interface on your Windows machine.

Prerequisites

- **Node.js**: Version 18.0.0 or higher
- **npm**: Version 9.0.0 or higher
- **API Server**: Must be running on `http://127.0.0.1:8000`

Check Node.js and npm versions

```
node --version
npm --version
```

If you don't have Node.js installed or need to update:
1. Download from nodejs.org (https://nodejs.org/)
2. Install the LTS (Long Term Support) version
3. Restart your terminal/PowerShell after installation

Setup Steps

1. Navigate to the Web Directory

```
cd C:\Users\Suxow\github_repos\SSVproff
cd web
```

2. Install Dependencies

This will install all required packages (React, Next.js, TypeScript, etc.):

```
npm install
```

Note: This may take 2-5 minutes depending on your internet connection.

3. Configure Environment Variables

A `.env.local` file has already been created with the correct configuration:

```
NEXT_PUBLIC_API_URL=http://127.0.0.1:8000/api/v1
```

Important: Make sure your API server is running on `http://127.0.0.1:8000` before starting the web interface.

4. Verify API Server is Running

Before starting the web interface, confirm your API server is running:

```
# In a separate PowerShell window, navigate to the API directory
cd C:\Users\Suxow\github_repos\SSVproff\api

# Activate virtual environment
..\venv\Scripts\Activate.ps1

# Start the server if not already running
uvicorn app.main:app --reload
```

You should see:

```
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
```

Running the Web Interface

Development Mode (Recommended for Testing)

```
npm run dev
```

This will start the Next.js development server with hot-reloading enabled.

Expected Output:

```
ready - started server on 0.0.0.0:3000, url: http://localhost:3000
info  - Loaded env from C:\Users\Suxow\github_repos\SSVproff\web\.env.local
```

Access the Application

Open your browser and navigate to:

```
http://localhost:3000
```

Available Features

Once the web interface is running, you can:

1. **Register a new account** - Navigate to `/register`
2. **Login** - Navigate to `/login`
3. **View Dashboard** - Navigate to `/dashboard` (requires authentication)
4. **View Profile** - Navigate to `/profile` (requires authentication)

Testing the Connection

Test API Connection

1. Open your browser's Developer Tools (F12)

2. Go to the Network tab

3. Navigate to the login page

4. Try to login or register

5. Check the Network tab for API requests to `http://127.0.0.1:8000/api/v1/...`

Expected API Endpoints

- **Register**: `POST http://127.0.0.1:8000/api/v1/auth/register`

- **Login**: `POST http://127.0.0.1:8000/api/v1/auth/login`

- **Get User**: `GET http://127.0.0.1:8000/api/v1/auth/me`

- **Tasks**: `GET/POST/PUT/DELETE http://127.0.0.1:8000/api/v1/tasks/...`

Troubleshooting

Error: "Cannot connect to API"

Cause: API server is not running or is running on a different port.

Solution:
1. Check if the API server is running
2. Verify it's running on `http://127.0.0.1:8000`
3. Check the `.env.local` file has the correct `NEXT_PUBLIC_API_URL`

Error: "EADDRINUSE: address already in use :::3000"

Cause: Port 3000 is already in use by another application.

Solution:

```
# Stop the process using port 3000 or run on a different port
npm run dev -- -p 3001
```

Then access the app at `http://localhost:3001`

Error: "npm: command not found" or "node: command not found"

Cause: Node.js is not installed or not in PATH.

Solution:
1. Install Node.js from nodejs.org (https://nodejs.org/)
2. Restart your terminal/PowerShell
3. Verify installation with `node --version` and `npm --version`

Changes not appearing

Cause: Browser cache or Next.js cache.

Solution:

```
# Clear Next.js cache
Remove-Item -Recurse -Force .next

# Restart dev server
npm run dev
```

Also try:

- Hard refresh in browser: `Ctrl + Shift + R`

- Clear browser cache

- Use incognito/private browsing mode

Database errors in API

Cause: Database not initialized or migrations not run.

Solution:

```
# In the API directory
cd C:\Users\Suxow\github_repos\SSVproff\api

# Activate virtual environment
..\venv\Scripts\Activate.ps1

# Run database initialization
python init_db.py
```

Production Build (Optional)

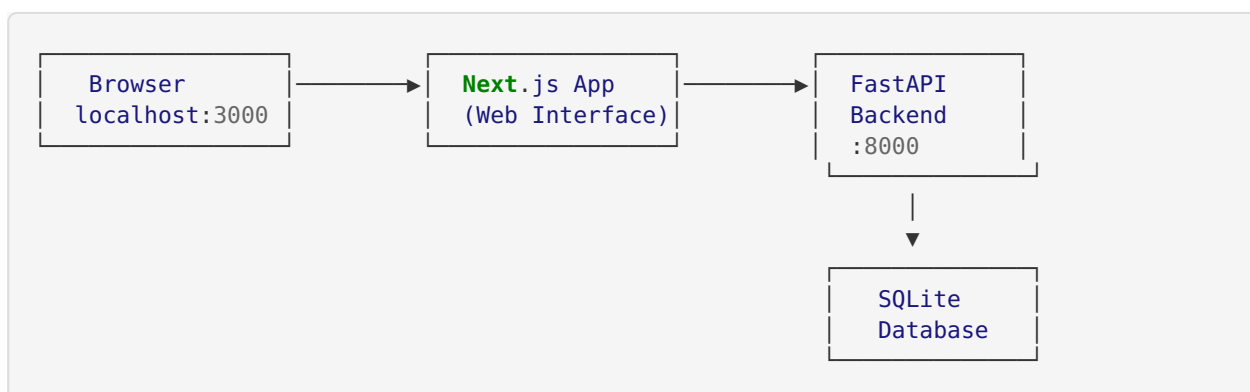To create a production build:

```
npm run build
npm run start
```

This will create an optimized build in the `.next` directory.

Available Scripts

- `npm run dev` - Start development server on port 3000
- `npm run build` - Create production build
- `npm run start` - Start production server
- `npm run lint` - Run ESLint to check code quality
- `npm run lint:fix` - Fix ESLint errors automatically
- `npm run format` - Format code with Prettier
- `npm run type-check` - Check TypeScript types
- `npm run test` - Run tests
- `npm run test:watch` - Run tests in watch mode

Architecture

```
┌─────────────────┐     ┌─────────────────┐     ┌─────────────────┐
│    Browser      │ ──▶ │   Next.js App   │ ──▶ │    FastAPI      │
│ localhost:3000  │     │ (Web Interface) │     │    Backend      │
└─────────────────┘     └─────────────────┘     │     :8000       │
                                                └─────────────────┘
                                                        │
                                                        ▼
                                                ┌─────────────────┐
                                                │     SQLite       │
                                                │    Database      │
                                                └─────────────────┘
```

Key Files

- `package.json` - Dependencies and scripts
- `.env.local` - Local environment configuration (not committed to git)
- `.env.example` - Example environment configuration (committed to git)
- `next.config.js` - Next.js configuration
- `src/lib/api.ts` - API client with backend integration
- `src/lib/auth.ts` - Authentication utilities
- `src/pages/` - Application pages (routing)
- `src/components/` - Reusable React components
- `src/contexts/` - React contexts (e.g., AuthContext)

Next Steps

1. ✅ Install dependencies: `npm install`
2. ✅ Verify API server is running
3. ✅ Start web interface: `npm run dev`
4. ✅ Open browser: `http://localhost:3000`
5. ✅ Register a new user
6. ✅ Test authentication and features

Support

If you encounter any issues:

1. Check the console output in your terminal
2. Check browser developer tools (F12) for errors
3. Verify API server is running and accessible
4. Check the `.env.local` configuration
5. Try clearing cache and restarting both servers

Last Updated: October 16, 2025