SSVproff Complete Setup Guide

This guide will help you set up and run the SSVproff application on your machine.

📋 Prerequisites

- **Python 3.8 or higher** - Download Python (https://www.python.org/downloads/)
- **Git** - Download Git (https://git-scm.com/downloads/)
- **pip** (usually comes with Python)
- A text editor or IDE (VS Code, PyCharm, etc.)

🚀 Step-by-Step Setup

1. Clone the Repository

If you haven't already cloned the repository:

```
git clone https://github.com/Serg2206/SSVproff.git
cd SSVproff
```

2. Checkout the Latest Branch

```
git fetch origin
git checkout feat/comprehensive-config-no-workflows
```

Or pull the latest changes if you're already on the branch:

```
git pull origin feat/comprehensive-config-no-workflows
```

3. Set Up the API (Backend)

Navigate to the API directory:

```
cd api
```

Create a virtual environment (recommended):

On Windows:

```
python -m venv venv
.\venv\Scripts\activate
```

On macOS/Linux:

```
python3 -m venv venv
source venv/bin/activate
```

Install dependencies:

```
pip install -r requirements.txt
```

Initialize the database:

```
python init_db.py
```

This will create:
- Database tables
- A default admin user:
- **Username:** `admin`
- **Password:** `admin123`

⚠️ **Important:** Change the admin password after first login!

Start the API server:

```
uvicorn app.main:app --reload --host 127.0.0.1 --port 8000
```

The API will be available at:
- **API:** http://127.0.0.1:8000
- **API Documentation:** http://127.0.0.1:8000/docs
- **Alternative Docs:** http://127.0.0.1:8000/redoc

4. Test the API

Open your browser and navigate to http://127.0.0.1:8000/docs

You should see the interactive API documentation (Swagger UI).

Try the following:

1. **Health Check:**
   - Go to `/health` endpoint
   - Click “Try it out”
   - Click “Execute”
   - You should see: `{"status": "healthy", ...}`

2. **Login:**
   - Go to `/api/v1/auth/login` endpoint
   - Click “Try it out”
   - Enter:
     - username: `admin`
     - password: `admin123`
     - Click “Execute”
     - Copy the `access_token` from the response

3. **Authorize:**
   - Click the “Authorize” button at the top
   - Enter: `Bearer <your_access_token>`
   - Click “Authorize”

4. **Create a Project:**
   - Go to `/api/v1/projects` POST endpoint
   - Click “Try it out”
   - Enter project details
   - Click “Execute”

🔧 Configuration

Environment Variables

The API uses a `.env` file for configuration. A default one is already created for you, but you can customize it:

```
# Edit the .env file
nano .env  # or use your preferred editor
```

Key configurations:
- `DATABASE_URL` - Database connection string
- `SECRET_KEY` - JWT secret key (change in production!)
- `CORS_ORIGINS` - Allowed origins for CORS
- `ACCESS_TOKEN_EXPIRE_MINUTES` - Token expiration time

Database Options

SQLite (Default):
- No additional setup required
- Perfect for development
- Database file: `ssvproff.db`

PostgreSQL (Production):
1. Install PostgreSQL
2. Create database: `CREATE DATABASE ssvproff;`
3. Update `.env`:
`DATABASE_URL=postgresql://user:password@localhost:5432/ssvproff`
4. Re-run `python init_db.py`

📁 Project Structure

```
SSVproff/
├── api/                        # Backend API
│   ├── app/
│   │   ├── main.py             # FastAPI application
│   │   ├── config.py           # Configuration
│   │   ├── database.py         # Database connection
│   │   ├── models.py           # Database models
│   │   ├── schemas.py          # Pydantic schemas
│   │   ├── auth.py             # Authentication
│   │   └── routers/            # API endpoints
│   │       ├── auth.py         # Auth routes
│   │       └── projects.py     # Project routes
│   ├── init_db.py              # DB initialization
│   ├── requirements.txt        # Python dependencies
│   ├── .env                    # Environment variables
│   └── README.md               # API documentation
├── web/                        # Frontend (Next.js)
├── docs/                       # Documentation
└── SETUP_GUIDE.md              # This file
```

🐛 Troubleshooting

Issue: “Module not found” errors

Solution:

```
# Make sure you're in the api directory
cd api

# Make sure virtual environment is activated
# Windows:
.\venv\Scripts\activate
# macOS/Linux:
source venv/bin/activate

# Reinstall dependencies
pip install -r requirements.txt
```

Issue: “Could not import module ‘main’“

Solution:
Make sure you’re running uvicorn with the correct module path:

```
# From the api directory:
uvicorn app.main:app --reload
```

Issue: Database errors

Solution:

```
# Delete the old database
rm ssvproff.db

# Reinitialize
python init_db.py
```

Issue: Port already in use

Solution:

```
# Use a different port
uvicorn app.main:app --reload --port 8001
```

Issue: Import errors with pydantic

Solution:
The code uses Pydantic v2. Make sure you have the correct version:

```
pip install --upgrade pydantic pydantic-settings
```

🔐 Security Notes

1. **Change the admin password** immediately after first login
2. **Update SECRET_KEY** in `.env` for production
3. **Use HTTPS** in production
4. **Never commit** `.env` file with real credentials
5. **Use PostgreSQL** for production (not SQLite)

📚 API Endpoints Reference

Authentication

- `POST /api/v1/auth/register` - Register new user
- `POST /api/v1/auth/login` - Login and get token
- `GET /api/v1/auth/me` - Get current user

Projects

- `GET /api/v1/projects` - List projects
- `POST /api/v1/projects` - Create project
- `GET /api/v1/projects/{id}` - Get project
- `PUT /api/v1/projects/{id}` - Update project
- `DELETE /api/v1/projects/{id}` - Delete project

Health

- `GET /health` - Health check

📞 Getting Help

If you encounter issues:

1. Check the logs in the terminal where uvicorn is running
2. Review the API documentation at http://127.0.0.1:8000/docs
3. Check the GitHub issues page
4. Make sure all dependencies are installed correctly

🎉 Next Steps

Once the API is running:

1. Explore the API documentation
2. Create test users and projects
3. Set up the web frontend (Next.js)
4. Customize the configuration for your needs
5. Read the detailed documentation in `/docs`

💡 Tips

- Use the interactive API docs (`/docs`) to test endpoints
- Keep the terminal open to see real-time logs
- Use `--reload` flag during development for auto-reload
- The database file `ssvproff.db` will be created in the `api` directory
- You can reset the database by deleting `ssvproff.db` and running `init_db.py` again

---

Happy coding! 🚀