Quick Start Guide for Windows

This guide will help you get the SSVproff API running on your Windows machine in just a few steps.

🔄 Pull the Latest Changes

First, make sure you have the latest code:

```
# Navigate to your repository
cd C:\Users\Suxow\github_repos\SSVproff

# Pull the latest changes
git pull origin feat/comprehensive-config-no-workflows
```

📋 What's New

The repository now has **TWO** working implementations:

1. Simple Implementation (Easy to start with)

Located in the root `api/` directory with straightforward structure:
- `api/auth.py` - JWT authentication
- `api/config.py` - Simple configuration
- `api/database.py` - Database connection
- `api/models.py` - Database models (User, Project, Dataset, Experiment)
- `api/schemas.py` - Pydantic schemas
- `api/routers/` - API endpoints
- `api/init_db.py` - Database initialization script

2. Advanced Implementation (Production-ready)

Located in `api/app/` with enterprise structure:
- `api/app/core/` - Configuration and security
- `api/app/db/` - Database session management
- `api/app/models/` - Database models (User, Task)
- `api/app/schemas/` - Pydantic schemas
- `api/app/api/v1/endpoints/` - API endpoints
- `api/scripts/init_db.py` - Advanced database initialization
- `api/alembic/` - Database migrations
- `api/tests/` - Comprehensive test suite
- Docker support

🚀 Quick Start (Simple Implementation)

Step 1: Install Python Dependencies

```
# Navigate to the api directory
cd api

# Create a virtual environment
python -m venv venv

# Activate the virtual environment
.\venv\Scripts\activate

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

Step 2: Initialize the Database

```
# Run the initialization script
python init_db.py
```

This creates:
- Database tables for User, Project, Dataset, and Experiment
- A default admin user (username: `admin`, password: `admin123`)

Step 3: Start the API Server

```
# Start the server with auto-reload
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 Docs:** http://127.0.0.1:8000/docs
- **Alternative Docs:** http://127.0.0.1:8000/redoc

🎯 Testing the API

1. **Open your browser** and go to: http://127.0.0.1:8000/docs

2. **Login to get an access token:**
   - Find the `/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 in Swagger UI:**
   - Click the **"Authorize"** button at the top

   - Enter: `Bearer <your_access_token>`
   - Click “Authorize”

4. **Try creating a project:**
   - Go to `/api/v1/projects` POST endpoint
   - Click “Try it out”
   - Enter:
   `json`
   ```
       {
         "name": "My First Project",
         "description": "Testing the API"
       }
   ```
   - Click “Execute”

🔧 Advanced Setup (Production Implementation)

If you want to use the advanced implementation with migrations and tests:

Step 1: Install Dependencies

```
cd api
pip install -r requirements.txt
pip install -r requirements-dev.txt
```

Step 2: Set Environment Variables

Create or edit `api/.env`:

```
# Database
DATABASE_URL=sqlite:///./ssvproff.db

# Security
SECRET_KEY=your-secret-key-here
ALGORITHM=HS256
ACCESS_TOKEN_EXPIRE_MINUTES=30

# API
PROJECT_NAME=SSVproff API
VERSION=0.1.0
API_V1_PREFIX=/api/v1
```

Step 3: Run Database Migrations

```
# Initialize the database with Alembic
alembic upgrade head

# Or use the init script
python scripts/init_db.py
```

Step 4: Start the Server

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

Step 5: Run Tests

```
pytest
```

📁 Project Structure

```
SSVproff/
├── api/
│   ├── app/                     # Main application
│   │   ├── api/                 # API routes (advanced)
│   │   ├── core/                # Config & security
│   │   ├── db/                  # Database (advanced)
│   │   ├── models/              # DB models (advanced)
│   │   ├── schemas/             # Pydantic schemas (advanced)
│   │   ├── services/            # Business logic
│   │   ├── auth.py              # Simple auth
│   │   ├── config.py            # Simple config
│   │   ├── database.py          # Simple DB
│   │   ├── models.py            # Simple models
│   │   ├── schemas.py           # Simple schemas
│   │   └── routers/             # Simple routes
│   ├── alembic/                 # Database migrations
│   ├── tests/                   # Test suite
│   ├── scripts/                 # Utility scripts
│   ├── init_db.py               # Simple DB init
│   ├── requirements.txt         # Dependencies
│   └── .env                     # Configuration
├── web/                         # Next.js frontend
├── docs/                        # Documentation
├── SETUP_GUIDE.md               # Detailed setup guide
└── QUICK_START_WINDOWS.md       # This file
```

🐛 Troubleshooting

Issue: "Module not found" error

Solution:

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

# Activate virtual environment
.\venv\Scripts\activate

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

Issue: "Could not import module 'main'"

Solution:
Make sure you're running the command from the `api` directory:

```
cd api
uvicorn app.main:app --reload
```

Issue: “Database is locked” or database errors

Solution:

```
# Delete the database file
Remove-Item ssvproff.db

# Reinitialize
python init_db.py
```

Issue: Port 8000 already in use

Solution:

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

📚 Available Endpoints

Authentication (Simple Implementation)

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

Projects (Simple Implementation)

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

Authentication (Advanced Implementation)

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

Tasks (Advanced Implementation)

- `GET /api/v1/tasks` - List all tasks
- `POST /api/v1/tasks` - Create a task
- `GET /api/v1/tasks/{id}` - Get task details
- `PUT /api/v1/tasks/{id}` - Update a task
- `DELETE /api/v1/tasks/{id}` - Delete a task

Health Check

- `GET /health` - Check API health status

💡 Tips

1. **Use the interactive API documentation** at `/docs` - it’s the easiest way to test the API

2. **Keep the terminal open** to see real-time logs
3. **The database file** `ssvproff.db` is created in the `api` directory
4. **To reset the database:** Delete `ssvproff.db` and run `python init_db.py` again
5. **For production:** Use PostgreSQL instead of SQLite

🔐 Security Notes

1. **Change the default admin password** immediately after first login!
2. **Update the SECRET_KEY** in `.env` before deploying to production
3. **Never commit** the `.env` file to Git (it's already in `.gitignore`)
4. **Use HTTPS** in production environments

📞 Need Help?

- **API Documentation:** http://127.0.0.1:8000/docs (when running)
- **Comprehensive Setup Guide:** See `SETUP_GUIDE.md`
- **Authentication Setup:** See `api/docs/AUTHENTICATION.md`
- **Testing Report:** See `TESTING_REPORT.md`

🎉 Next Steps

1. **Explore the API** using the interactive documentation
2. **Create test users and projects**
3. **Try the web frontend** (see `web/` directory)
4. **Read the detailed documentation** in the `docs/` folder
5. **Run the tests** to ensure everything works correctly

Happy coding! 🚀

If you encounter any issues, check the troubleshooting section or refer to the detailed `SETUP_GUIDE.md`.