Complete Project Setup Guide - S3 Storage System

Prerequisites & Software Installation

1. Required Software

- **Python 3.12** ✅ (You already have this)
- **Node.js** (Latest LTS version - 18.x or 20.x)
- **npm** (Comes with Node.js)
- **Docker Desktop** (Optional - only if you want to use Docker)
- **Git** (For version control)

2. Download & Install Node.js

1. Go to nodejs.org
2. Download the **LTS version** (Long Term Support)
3. Install it (npm comes automatically with Node.js)
4. Verify installation:

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

3. Docker Desktop (Optional)

- Download from docker.com
- Install and start Docker Desktop
- You'll see Docker icon in system tray when running

Project Setup Process

Step 1: Project Directory Setup

```bash
# Navigate to your desired location (e.g., Desktop)
cd C:\Users\YourUsername\Desktop

# Clone or create project directory
mkdir simple-s3-storage
cd simple-s3-storage
```

Step 2: Create Project Structure

Create the folder structure as shown in your paste.txt file, or if you have the code, extract it to this directory.

Backend Setup (Python/Flask)

Step 3: Backend Environment Setup

Directory: `simple-s3-storage/backend/`

```bash
# Navigate to backend directory
cd backend

# Create Python virtual environment
python -m venv venv

# Activate virtual environment
# On Windows (Command Prompt):
venv\Scripts\activate
# On Windows (PowerShell):
venv\Scripts\Activate.ps1
# On macOS/Linux:
source venv/bin/activate

# You should see (venv) in your command prompt now
```

Step 4: Install Python Dependencies

Directory: `simple-s3-storage/backend/` (with venv activated)

```bash
```

```
# Install required packages
pip install flask
pip install flask-cors
pip install bcrypt
pip install werkzeug
pip install python-dotenv
pip install requests

# Or if you have requirements.txt:
pip install -r requirements.txt

# Verify installation
pip list
```

Step 5: Create Data Directories

Directory: `simple-s3-storage/backend/`

```
bash

# Create necessary directories
mkdir ../data
mkdir ../data/uploads
mkdir ../data/metadata

# Create initial JSON files
echo {} > ../data/users.json
echo {} > ../data/permissions.json
echo {} > ../data/search_index.json
echo {} > ../data/metadata/file_metadata.json
```

Step 6: Run Backend Server

Directory: `simple-s3-storage/backend/` (with venv activated)

```
bash

# Run the Flask application
python app.py

# You should see:
# * Running on http://127.0.0.1:5000
# * Debug mode: on/off
```

Keep this terminal open - the backend server needs to keep running.

Frontend Setup (React)

Step 7: Frontend Environment Setup

Open a NEW terminal/command prompt

Directory: `simple-s3-storage/frontend/`

```bash
# Navigate to frontend directory
cd C:\Users\YourUsername\Desktop\simple-s3-storage\frontend

# Initialize npm project (if package.json doesn't exist)
npm init -y

# Install React and required dependencies
npm install react react-dom react-scripts
npm install axios
npm install react-router-dom
npm install @tailwindcss/forms

# Install Tailwind CSS
npm install -D tailwindcss postcss autoprefixer
npx tailwindcss init -p
```

Step 8: Configure Tailwind CSS

Directory: `simple-s3-storage/frontend/`

Create or update `tailwind.config.js`:

```javascript
module.exports = {
  content: [
    "./src/**/*.{js,jsx,ts,tsx}",
    "./public/index.html"
  ],
  theme: {
    extend: {},
  },
  plugins: [
    require('@tailwindcss/forms'),
  ],
}
```

Step 9: Run Frontend Development Server

Directory: `simple-s3-storage/frontend/`

```bash
# Start the React development server
npm start

# You should see:
# Local:            http://localhost:3000
# On Your Network:  http://192.168.x.x:3000
```

Your browser should automatically open to `http://localhost:3000`

Running Both Servers

You Need TWO Terminal Windows:

Terminal 1 - Backend Server

```bash
cd C:\Users\YourUsername\Desktop\simple-s3-storage\backend
venv\Scripts\activate  # Activate virtual environment
python app.py          # Runs on http://localhost:5000
```

Terminal 2 - Frontend Server

```bash
cd C:\Users\YourUsername\Desktop\simple-s3-storage\frontend
npm start              # Runs on http://localhost:3000
```

Both servers must be running simultaneously!

PyCharm Integration

Step 10: PyCharm Setup

1. **Open PyCharm**
2. **Open Project:** File → Open → Select `simple-s3-storage` folder
3. **Configure Python Interpreter:**
   - File → Settings → Project → Python Interpreter
   - Click gear icon → Add → Existing Environment

- Navigate to: `simple-s3-storage/backend/venv/Scripts/python.exe`
- Click OK

4. **Run Configuration:**
   - Right-click `backend/app.py` → Run 'app'
   - Or create run configuration: Run → Edit Configurations → + → Python Script
   - Script path: `/path/to/backend/app.py`
   - Working directory: `/path/to/backend/`

Docker Setup (Alternative Method)

Step 11: Docker Deployment (Optional)

Only if you want to use Docker instead of manual setup:

Directory: `simple-s3-storage/`

```bash
# Make sure Docker Desktop is running
docker --version

# Build and run with Docker Compose
docker-compose up --build

# This will:
# - Build both frontend and backend
# - Start both services
# - Frontend: http://localhost:3000
# - Backend: http://localhost:5000
```

Note: If using Docker, you don't need the manual Python/Node setup above.

Troubleshooting

Common Issues & Solutions:

Backend Issues:

```bash
```

```
# If "python" command not found:
python3 app.py

# If virtual environment issues:
python -m pip install virtualenv
python -m virtualenv venv

# If port 5000 is busy:
# Edit app.py and change: app.run(port=5001)
```

Frontend Issues:

```
bash

# If npm install fails:
npm cache clean --force
npm install

# If port 3000 is busy:
# Create .env file in frontend folder:
echo "PORT=3001" > .env

# If Tailwind CSS not working:
npm run build:css
```

Permission Issues:

```
bash

# Windows PowerShell execution policy:
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser

# File permission issues:
chmod +x deploy.sh  # On macOS/Linux
```

Testing Your Setup

Step 12: Verify Everything Works

1. **Backend Test:** Visit `http://localhost:5000` - should show API response
2. **Frontend Test:** Visit `http://localhost:3000` - should show login page
3. **Create Account:** Register with email/password
4. **Upload Test:** Create bucket and upload a file

5. **Search Test:** Search for uploaded files

Development Workflow

Daily Development:

1. **Start Backend:**

```bash
cd backend
venv\Scripts\activate
python app.py
```

2. **Start Frontend:**

```bash
cd frontend
npm start
```

3. **Make Changes:** Edit files in PyCharm
4. **Auto-Reload:** Both servers auto-reload on file changes

Production Deployment:

```bash
# Build frontend for production
cd frontend
npm run build

# Deploy to server or cloud platform
```

Directory Summary

```
simple-s3-storage/
├── backend/        # Run: python app.py
├── frontend/       # Run: npm start
├── data/           # Created automatically
├── docker/         # Run: docker-compose up
└── README.md
```

Key Commands Reference:

- **Backend:** `cd backend && python app.py`

- **Frontend:** `cd frontend && npm start`
- **Docker:** `docker-compose up`
- **Install Backend:** `cd backend && pip install -r requirements.txt`
- **Install Frontend:** `cd frontend && npm install`

You're all set! Both servers should be running and your S3 storage system should be accessible at `http://localhost:3000`.