LLMscope Project Structure & Setup Guide

Phase 2 - Directory Organization

Root Directory Structure

```
LLMscope/
│
├── 📦 DEPLOYABLE (What users download)
│   ├── docker-compose.yml          ⭐ Main entry point
│   ├── .env.example                ⭐ Config template
│   ├── README.md                   ⭐ Quick start
│   ├── Dockerfile.backend
│   ├── Dockerfile.frontend
│   ├── nginx.conf
│   ├── requirements.txt
│   │
│   ├── frontend/
│   │   ├── package.json
│   │   ├── package-lock.json
│   │   ├── vite.config.ts
│   │   ├── tailwind.config.js
│   │   ├── postcss.config.js
│   │   ├── tsconfig.json
│   │   ├── index.html
│   │   └── src/
│   │       ├── main.tsx            ✅ UPDATED
│   │       ├── index.css
│   │       ├── SetupWizard.jsx     ✅ NEW
│   │       ├── Dashboard_ollama_revB.jsx  ✅ UPDATED
│   │       └── SPCChart_ollama_revB.jsx
│   │
│   ├── backend/
│   │   ├── app.py                  ✅ UPDATED (Phase 2)
│   │   ├── monitor_apis.py
│   │   └── data/
│   │       └── llmscope.db         (auto-created)
│   │
│   └── tests/
│       ├── pytest.ini              ✅ NEW
│       ├── conftest.py             ✅ NEW
│       ├── test_e2e_phase2.py      ✅ NEW
```

```
│   └── fixtures/
│       └── mock_telemetry.json
│
├── 📋 DOCUMENTATION (Git tracked)
│   ├── docs/
│   │   ├── INSTALL.md              ⭐ Non-IT setup guide
│   │   ├── QUICKSTART.md           ⭐ 5-minute start
│   │   ├── ARCHITECTURE.md
│   │   ├── API.md
│   │   ├── TROUBLESHOOTING.md
│   │   ├── EMAIL_SETUP.md          (Gmail, SendGrid, etc.)
│   │   ├── SLACK_SETUP.md          (Webhook instructions)
│   │   └── DEVELOPMENT.md          (For developers)
│   │
│   └── .github/
│       └── workflows/
│           ├── e2e-tests.yml       ✅ NEW (CI/CD)
│           └── build.yml
│
├── 🗃️ ARCHIVE (Git ignored - your internal stuff)
│   ├── marketing/
│   │   ├── screenshots/
│   │   ├── demo-videos/
│   │   ├── landing-page/
│   │   └── pitch-deck/
│   │
│   ├── financials/
│   │   ├── pricing-models/
│   │   ├── cost-analysis/
│   │   └── projections/
│   │
│   ├── business/
│   │   ├── roadmap/
│   │   ├── meeting-notes/
│   │   └── customer-feedback/
│   │
│   ├── legal/
│   │   ├── licenses/
│   │   ├── terms-of-service/
│   │   └── privacy-policy/
│   │
│   └── internal/
│       ├── brainstorm/
│       ├── experiments/
```

```
│   └── scratch-work/
│
├── 📝 VERSION CONTROL
│   ├── .gitignore              ✅ UPDATED
│   ├── .gitattributes
│   ├── LICENSE
│   ├── CHANGELOG.md
│   └── VERSION                 (0.2.0)
│
└── 🚀 DEPLOYMENT
    ├── docker-compose.prod.yml     (for production)
    └── .env.prod.example           (prod config template)
```

.gitignore (Updated for Archive)

```
gitignore
```

```
# Archive directory (internal business files - NOT tracked)
/archive/
/archive/**

# Environment variables (secrets)
.env
.env.local
.env.*.local

# Python
__pycache__/
*.py[cod]
*$py.class
*.so
.Python
build/
develop-eggs/
dist/
downloads/
eggs/
.eggs/
lib/
lib64/
parts/
sdist/
var/
wheels/
*.egg-info/
.installed.cfg
*.egg
.pytest_cache/
.coverage
htmlcov/

# Node/Frontend
node_modules/
dist/
.DS_Store
*.tsbuildinfo

# IDE
.vscode/
.idea/
```

```
*.swp
*.swo
*~
.project
.classpath

# Database (local development)
data/llmscope.db
*.db

# Docker
docker-compose.override.yml

# OS
.DS_Store
Thumbs.db
```

Migration Steps (For You)

Step 1: Create Directory Structure

bash

```
# Create archive directory
mkdir -p archive/{marketing,financials,business,legal,internal}

# Move your existing files
# (adjust paths based on your current layout)
mv ./marketing/* ./archive/marketing/
mv ./financials/* ./archive/financials/
# etc...

# Create documentation directory
mkdir -p docs
mkdir -p .github/workflows
mkdir -p tests/fixtures
```

Step 2: Copy Updated Files

bash

```
# Backend
cp app.py ./backend/

# Frontend
cp SetupWizard.jsx ./frontend/src/
cp Dashboard_ollama_revB.jsx ./frontend/src/
cp main.tsx ./frontend/src/

# Tests (we'll create these next)
# cp test_e2e_phase2.py ./tests/
# cp conftest.py ./tests/
```

Step 3: Create .env.example

bash

```
# Copy as template (don't commit actual .env)
cp .env .env.example
git rm --cached .env  # Stop tracking actual .env
```

Step 4: Commit Clean Structure

bash

```
git add .
git commit -m "Phase 2: Clean project structure with archive directory"
```

What Goes in Archive/ (Ignored)

✅ **DO PUT IN `/archive`:**

- Marketing materials & screenshots
- Financial spreadsheets & projections
- Business meeting notes
- Customer feedback
- Pricing strategy docs
- Legal documents
- Internal brainstorming

- Experimental code
- TODO lists specific to you
- Pitch decks
- Demo videos

❌ **DON'T PUT IN `/archive`:**

- Source code
- Configuration files
- Dependencies (requirements.txt, package.json)
- Tests
- Documentation meant for users
- Deployment configs

Clean Repo Benefits

✅ Users clone repo → only see what they need
✅ No financial data exposed
✅ No marketing clutter
✅ Professional appearance
✅ Easy to distribute
✅ Your internal stuff stays private
✅ CI/CD doesn't get confused

Next Steps

1. **Organize your files** using structure above
2. **Create `.gitignore`** with archive directory
3. **I'll create:**
   - E2E test suite
   - docker-compose.yml for users
   - .env.example template
   - INSTALL.md for non-IT users

Sound good? 🚀