SecureMed Installation Guide

Version: 1.0

Release Date: Fall 2025

Project: SecureMed Healthcare Cybersecurity & HIPAA Compliance Platform

Institution: Florida International University, Knight Foundation School of Computing and Information Sciences

---

1.0 Introduction

This installation guide provides step-by-step instructions for setting up and launching the SecureMed Healthcare Cybersecurity & HIPAA Compliance Platform. It ensures consistent installation across Windows, macOS, and Linux environments.

The system uses:

- **Backend**: Python Flask REST API
- **Frontend**: React-based user interface
- **Database**: SQLite with Fernet AES-128 encryption
- **Security**: HTTPS-ready, JWT authentication-ready, role-based access control

This guide aligns with deliverables from Sprint 5 and Sprint 6 (documentation and submission packaging phases).

2.0 System Requirements

2.1 Software Requirements

Requirement	Version	Notes
Python	3.8+	Required for Flask backend
pip	Latest	Python package manager
Virtual Environment	Built-in (venv)	Included with Python 3.3+
Web Browser	Modern (2020+)	Chrome, Firefox, Edge, or Safari
Git (optional)	Latest	For version control
Text Editor/IDE	Any	VS Code recommended

2.2 Hardware Requirements

Component	Minimum	Recommended
RAM	4 GB	8 GB
Disk Space	200 MB	500 MB
Processor	Any modern CPU	2+ cores
OS	Windows, macOS, Linux	Any 64-bit system

2.3 Network Requirements

- Local network access only (no internet required)
- Port 5000 available (default Flask port)
- Localhost (127.0.0.1) access

3.0 Directory Structure

Your project folder should contain the following structure:

```
Cap_Finaldev/
├── webapp.py                    # Main Flask application
├── generate_report.py           # PDF report generation
├── requirements.txt             # Python dependencies
├── securemed.db                 # SQLite database (auto-created)
├── templates/                   # HTML templates with React
│   ├── login.html
│   ├── dashboard_react.html     # Admin dashboard
│   ├── user_dashboard_react.html # Nurse dashboard
│   ├── edr.html                 # EDR security panel
│   ├── training_simulator.html  # HIPAA training
│   ├── patients.html
│   ├── audit_trail.html
│   └── ...
├── static/                      # CSS, JavaScript, images
│   ├── style.css
│   ├── session_timeout.js
│   └── ...
├── docs/                        # Documentation
│   ├── INSTALL.md              # Installation guide
│   ├── HOW_TO_USE.md           # User manual
│   ├── FEATURES.md             # Feature documentation
│   ├── TESTING.md              # Test suite guide
│   ├── TROUBLESHOOTING.md      # Common issues
│   ├── FINAL_REPORT.md         # Project report
│   └── TEAM_CONTRIBUTIONS.md   # Team member breakdown
├── test_webapp.py               # Automated test suite
├── seed_*.py                    # Database seeding scripts
└── venv/                        # Virtual environment (created during setup)
```

This structure aligns with the final documentation package prepared in Sprints 5-6.

4.0 Installation Instructions

4.1 Windows Installation (Step-by-Step)

4.1.1 Open the Terminal

Option 1: VS Code

- Open VS Code
- Press `Ctrl + `` (backtick)
- Terminal will open at the bottom

Option 2: Command Prompt/PowerShell

- Press `Windows Key + R`
- Type `cmd` and press Enter

4.1.2 Navigate to Project Folder

```
cd path\to\Cap_Finaldev
```

Example:

```
cd C:\Users\YourName\Documents\Cap_Finaldev
```

4.1.3 Create Virtual Environment

```
python -m venv venv
```

This creates a `venv/` folder with isolated Python packages.

4.1.4 Activate Virtual Environment

```
venv\Scripts\activate
```

Success indicator: Your prompt will show `(venv)` at the beginning.

If you get an execution policy error:

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
```

Then try the activate command again.

4.1.5 Install Dependencies

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

This installs all required packages:

- Flask 3.1.2
- Flask-CORS 6.0.1
- Cryptography (Fernet)
- ReportLab (PDF generation)
- requests (for testing)

Expected output:

```
Successfully installed flask-3.1.2 cryptography-46.0.3 reportlab-4.4.4 ...
```

4.1.6 Launch the Application

```
python webapp.py
```

Expected output:

```
 * Running on http://127.0.0.1:5000
 * Press CTRL+C to quit
```

4.1.7 Access the Application

Open your web browser and navigate to:

```
http://127.0.0.1:5000/login
```

You should see the SecureMed login page.

4.2 macOS Installation (Step-by-Step)

4.2.1 Open Terminal

Press `Cmd + Space` to open Spotlight, type `Terminal`, and press Enter.

4.2.2 Navigate to Project Folder

```
cd /path/to/Cap_Finaldev
```

Example:

```
cd ~/Documents/Cap_Finaldev
```

4.2.3 Create Virtual Environment

```
python3 -m venv venv
```

Note: macOS may use `python3` instead of `python`

4.2.4 Activate Virtual Environment

```
source venv/bin/activate
```

Success indicator: Your prompt will show `(venv)` at the beginning.

4.2.5 Install Dependencies

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

4.2.6 Launch the Application

```
python3 webapp.py
```

Expected output:

```
 * Running on http://127.0.0.1:5000
 * Press CTRL+C to quit
```

4.2.7 Access the Application

Open your web browser and navigate to:

```
http://127.0.0.1:5000/login
```

4.3 Linux Installation (Step-by-Step)

4.3.1 Open Terminal

Press `Ctrl + Alt + T` (most Linux distributions).

4.3.2 Navigate to Project Folder

```
cd /path/to/Cap_Finaldev
```

4.3.3 Install Python (if needed)

```
sudo apt-get update
sudo apt-get install python3 python3-venv python3-pip
```

4.3.4 Create Virtual Environment

```
python3 -m venv venv
```

4.3.5 Activate Virtual Environment

```
source venv/bin/activate
```

4.3.6 Install Dependencies

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

4.3.7 Launch the Application

```
python3 webapp.py
```

4.3.8 Access the Application

Open your web browser and navigate to:

```
http://127.0.0.1:5000/login
```

5.0 Default Login Credentials

After successful installation, use these credentials to access the system:

Admin Account

Field	Value
Role	Administrator
Username	`admin`
Password	`Admin123!`

Admin capabilities:

- Create/manage user accounts
- View all violations and audit logs
- Generate compliance reports
- Simulate breach incidents
- Reset demo data
- Access EDR security panel

Demo User Accounts

Role	Username	Password	Purpose
Nurse	`stefan`	`Stefan123!`	Patient management, training
Nurse	`ana`	`Ana123!`	Patient access, assignments
Nurse	`jordan`	`Jordan123!`	Dashboard access, training
Nurse	`jeremiah`	`Jeremiah123!`	Testing account
Nurse	`mumin`	`Mumin123!`	Testing account

User capabilities:

- View assigned patients
- Complete HIPAA training
- Complete task assignments
- View personal compliance score
- View audit trail (own actions only)

6.0 Initial System Setup

6.1 Admin Quick Setup

After logging in as **admin**:

1. **Access Admin Dashboard**
    - Navigate to the main dashboard after login
2. **Click "⚡ Quick Setup" Button**
    - Located in the admin panel

- Generates demo data automatically

3. **System Generates**
    - 10-15 sample patient records
    - 5-10 sample assignments
    - 3-5 sample security vulnerabilities
    - 2-3 sample HIPAA violations
4. **Demo Data Ready**
    - You can now test the full system
    - Try the training simulator as a nurse user
    - Generate compliance reports
    - Simulate breach incidents

6.2 Manual Data Entry (Alternative)

If you prefer to add data manually:

1. Log in as **admin**
2. Go to **Patients** section
3. Click **"+ Add Patient"**
4. Fill in patient information:
    - First Name (required)
    - Last Name (required)
    - Date of Birth (required)
    - Email (optional)
    - Phone (optional)
    - Address (optional)
    - SSN (encrypted automatically)
5. Click **"Save Patient"**
6. Patient appears in the patient list

7.0 Verifying Installation

7.1 Check Flask Server Status

In the terminal where you ran `python webapp.py`, you should see:

```
 * Serving Flask app 'webapp'
 * Debug mode: on
 * Running on http://127.0.0.1:5000
 * Press CTRL+C to quit
```

7.2 Verify Database Creation

The SQLite database (`securemed.db`) is created automatically on first run. Check that it exists:

Windows:

```
dir securemed.db
```

macOS/Linux:

```
ls -la securemed.db
```

Expected output:

```
-rw-r--r-- 1 user group 45056 Dec 03 2025 securemed.db
```

7.3 Test Login

1. Open http://127.0.0.1:5000/login (http://127.0.0.1:5000/login)
2. Enter username: `admin`
3. Enter password: `Admin123!`
4. Click "Login"
5. You should see the admin dashboard

7.4 Run Automated Tests (Optional)

In a **new terminal** (while Flask is still running):

```
# Activate virtual environment if not already active
source venv/bin/activate  # macOS/Linux
# OR
venv\Scripts\activate     # Windows

# Run all tests
python3 -m pytest test_webapp.py -v
# OR
python test_webapp.py
```

Expected output:

```
Ran 34 tests in 0.523s
OK
```

8.0 Stopping the Application

8.1 Stop Flask Server

In the terminal where Flask is running:

```
Ctrl + C
```

Expected output:

```
KeyboardInterrupt
 * Running on http://127.0.0.1:5000
 * Press CTRL+C to quit
```

8.2 Deactivate Virtual Environment

After stopping Flask:

```
deactivate
```

Verification: The `(venv)` prefix disappears from your terminal prompt.

9.0 Common Installation Errors & Troubleshooting

9.1 "Flask Not Found" or "ModuleNotFoundError"

Error message:

```
ModuleNotFoundError: No module named 'flask'
```

Cause: Virtual environment not activated or dependencies not installed.

Fix:

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

# Reinstall dependencies
pip install -r requirements.txt

# Run Flask again
python webapp.py
```

9.2 "Port Already in Use"

Error message:

```
Address already in use
OSError: [Errno 48] Address already in use
```

Cause: Port 5000 is being used by another process.

Fix (macOS/Linux):

```
# Find process using port 5000
lsof -ti:5000

# Kill the process
lsof -ti:5000 | xargs kill -9

# Verify port is free
lsof -ti:5000  # Should return nothing
```

Fix (Windows):

```
# Find process using port 5000
netstat -ano | findstr :5000

# Kill the process (replace <PID> with actual process ID)
taskkill /PID <PID> /F

# Verify port is free
netstat -ano | findstr :5000  # Should return nothing
```

9.3 "Template Not Found" Error

Error message:

```
TemplateNotFound: login.html
```

Cause: Running Flask from wrong directory.

Fix:

```
# Verify you're in the correct directory
pwd                    # macOS/Linux
cd                     # Windows

# Should output: /path/to/Cap_Finaldev or C:\path\to\Cap_Finaldev

# If not, navigate to correct directory
cd /path/to/Cap_Finaldev

# Then run Flask
python webapp.py
```

9.4 "Python Not Found"

Error message:

```
'python' is not recognized as an internal or external command
```

Cause: Python not installed or not in system PATH.

Fix:

Windows:

1. Download Python from https://www.python.org/downloads/ (https://www.python.org/downloads/)
2. Run installer
3. **CHECK**: "Add Python to PATH" checkbox
4. Click "Install Now"
5. Restart terminal

macOS:

```
# Install via Homebrew
brew install python3

# Verify installation
python3 --version
```

Linux:

```
# Ubuntu/Debian
sudo apt-get install python3 python3-pip

# Verify installation
python3 --version
```

9.5 "Permission Denied" (macOS/Linux)

Error message:

```
Permission denied: './venv/bin/activate'
```

Cause: Virtual environment not properly created.

Fix:

```
# Remove virtual environment and recreate
rm -rf venv

# Create fresh virtual environment
python3 -m venv venv

# Activate (should work now)
source venv/bin/activate
```

9.6 "Database Locked" Error

Error message:

```
sqlite3.OperationalError: database is locked
```

Cause: Multiple Flask instances running or database write conflict.

Fix:

```
# Stop all Flask processes
Ctrl + C

# Wait 10 seconds
# If that doesn't work:

# macOS/Linux
lsof -i :5000 | grep LISTEN
kill -9 <PID>

# Windows
netstat -ano | findstr :5000
taskkill /PID <PID> /F

# Delete temporary database locks
rm securemed.db-journal  # macOS/Linux
del securemed.db-journal # Windows

# Restart Flask
python webapp.py
```

10.0 Next Steps After Installation

After successful installation:

1. **Read the User Guide**: Open `docs/HOW_TO_USE.md`
2. **Review Features**: Open `docs/FEATURES.md`
3. **Run Tests**: Execute `python test_webapp.py`
4. **Try Admin Functions**:
    - Quick Setup (generate demo data)
    - View EDR panel
    - Simulate a breach
    - Generate compliance report
5. **Try User Functions** (as nurse):
    - Complete HIPAA training
    - Add/edit patient records
    - View personal compliance score
    - Complete task assignments

11.0 Production Deployment Considerations

This installation guide is for **development/demo purposes**. For production deployment:

Required Changes

Component	Demo	Production
Web Server	Flask dev server	Gunicorn/uWSGI + nginx
Database	SQLite	PostgreSQL with encryption
HTTPS	HTTP only	Valid SSL/TLS certificate
Encryption Keys	Hardcoded	AWS KMS/HashiCorp Vault
Session Timeout	2 minutes	15-30 minutes
Multi-Factor Auth	Not implemented	TOTP/FIDO2 required
Rate Limiting	Not implemented	Flask-Limiter
Logging	Console	Centralized (Splunk/ELK)

For production deployment guidance, see `docs/FINAL_REPORT.md` — Deployment & Operations section.

12.0 Support & Troubleshooting

12.1 Check Documentation

- **Installation Issues**: See section 9.0 (Common Errors)
- **Usage Questions**: See `docs/HOW_TO_USE.md`
- **Feature Help**: See `docs/FEATURES.md`
- **Testing**: See `docs/TESTING.md`
- **General Issues**: See `docs/TROUBLESHOOTING.md`

12.2 System Information

When reporting issues, provide:

```
# Python version
python --version

# Flask version
pip show flask

# Operating System
# Windows: ver
# macOS: sw_vers
# Linux: lsb_release -a

# Virtual environment status
echo $VIRTUAL_ENV  # macOS/Linux
echo %VIRTUAL_ENV% # Windows
```

13.0 Uninstallation

To completely remove SecureMed:

13.1 Stop the Application

```
# In the running terminal
Ctrl + C

# Deactivate virtual environment
deactivate
```

13.2 Remove Virtual Environment

```
# macOS/Linux
rm -rf venv

# Windows
rmdir /s /q venv
```

13.3 Remove Database (Optional)

```
# macOS/Linux
rm securemed.db

# Windows
del securemed.db
```

13.4 Remove Project Folder (Optional)

```
# macOS/Linux
cd ..
rm -rf Cap_Finaldev

# Windows
cd ..
rmdir /s /q Cap_Finaldev
```

14.0 Conclusion

This installation guide provides all necessary steps to deploy SecureMed on Windows, macOS, or Linux. The system is now ready for:

- □ Demonstration and evaluation
- □ Cybersecurity capstone presentations
- □ HIPAA compliance training
- □ Educational use in healthcare security courses
- □ Foundation for production deployment

For questions or issues, refer to the comprehensive documentation in the `docs/` folder or consult the troubleshooting section above.

Document Information:

- **Version**: 1.0 - Final
- **Last Updated**: December 2025
- **Author**: SecureMed Team
- **Institution**: Florida International University
- **Course**: CIS 4914 - Cybersecurity Capstone Project II