

# TEST SCENARIO TS-01: SYSTEM INSTALLATION & LAUNCH

**Project Name:** Library Inventory Manager

**Document ID:** TS-01

**Module:** Infrastructure & Setup

**Author:** Dominik Hoch

**Date:** 2.1.2026

## 1. TEST OBJECTIVE

To verify that the application can be deployed on a clean environment. This includes setting up the Python environment, importing the database structure, configuring credentials, and successfully starting the web server.

## 2. PREREQUISITES

The testing machine must meet the following requirements:

- **OS:** Windows, macOS, or Linux.
  - **Software:** Python 3.9 (or newer), MySQL Server 8.0 (or newer).
  - **Source Code:** The project folder `library-inventory-manager` is available locally.
- 

## 3. TEST STEPS EXECUTION

Please execute the following steps in the given order.

### STEP 1: Obtain Source Code

- **Action:** Open a terminal/command prompt. Run the command to download the project: `git clone https://github.com/your-username/library-inventory-manager.git` Then navigate into the folder: `cd library-inventory-manager`
- **Expected Result:** The project repository is downloaded locally, and the terminal path is set to the project root directory.

## STEP 2: Environment Preparation

- **Action:** Inside the project root folder, run the command to create a virtual environment: `python -m venv .venv`
- **Expected Result:** A new folder named `.venv` is created in the project directory.

## STEP 3: Environment Activation

- **Action:** Activate the virtual environment:
  - *Windows:* `.venv\Scripts\activate`
  - *Mac/Linux:* `source .venv/bin/activate`
- **Expected Result:** The command line prompt changes to show `(.venv)`.

## STEP 4: Dependency Installation

- **Action:** Install required packages by running: `pip install -r requirements.txt`
- **Expected Result:** The system installs `Flask` and `mysql-connector-python`. The process finishes without any red error messages.

## STEP 5: Database Import

- **Action:** Import the SQL schema and data. Run command: `mysql -u root -p < sql/install.sql` (*Enter your MySQL root password when prompted*).
- **Expected Result:** The command completes silently or with no errors. The database `library_db` is created containing 5 tables (`books`, `authors`, `categories`, `book_authors`, `loans`) and 2 views.

## STEP 6: Configuration

- **Action:** Open `config/db_config.json` in a text editor. Update the fields `"user"` and `"password"` to match your local MySQL credentials. Save the file.
- **Expected Result:** The file is saved successfully with valid JSON syntax.

## STEP 7: System Launch

- **Action:** Start the application server: `python app.py`
- **Expected Result:** The console displays: `* Running on http://127.0.0.1:5000`. The application waits for connections and does not crash.

## STEP 8: UI Verification

- **Action:** Open a web browser and navigate to `http://127.0.0.1:5000`.
- **Expected Result:** The "Book List" page loads. The table is visible (containing demo data like "Válka s Mloky").

---

## 4. TROUBLESHOOTING NOTES

- **Error 1045 (Access Denied):** Verify that the password in `config/db_config.json` is correct.
  - **Error 1049 (Unknown Database):** Ensure Step 4 was executed correctly.
  - **ModuleNotFoundError:** Ensure the virtual environment is active (Step 2).
- 

## 5. TEST RESULT

*(To be filled by the Tester)*

**Status:** ☐ **PASS** (System started successfully) ☐ **FAIL** (System failed to start)

**Tester Name:** \_\_\_\_\_

**Signature:** \_\_\_\_\_

**Date:** \_\_\_\_\_