

# Documentation for further maintenance

By the HTML-Group

From the Bili Hackathon Wilhelmsburg  
2022

# Contents

1. Raspberry Pi environment
  - a. Install Pi
  - b. Install Environment
  - c. Database
  - d. Starting up
2. Local test environment
  - a. Requirements
  - b. Instructions for Setup
3. Application Insights
  - a. App
  - b. Data
  - c. SQLBuilder
  - d. Database.sql

## 1. Documentation of the runtime-environment for the prototype

- A. Installing the Raspberry Pi (further called RP)
  - A.1. Install the RP OS via "RaspberryPiImager" onto the SD-card
  - A.2. Plug the card into fully heat-sinked RP
  - A.3. First start the RP
  - A.4. Make sure Internet is connected and functional
    - A.4.1. Maybe the time settings must be corrected for the RP has no rtc
  - A.5. Install all Updates
- B. Install the Repository
  - B.1. Copy the full repository from the main-branch onto the RP
    - USB-Stick for example, or if access by clone via git
  - B.2. Setup the virtual environment for the modules flask and mysql.connector
- C. Get database ready
  - C.1. Install MySQL via the apt package manager and create a user called test with password test (IMPORTANT)
  - C.2. Setup the tables by copy-paste the database.sql text into the mysql-terminal
- D. Start the flask app
  - D.1. Start the app.py from within the repository
  - D.2. Go to 127.0.0.1:5000/

## 2. How to set up Docker environment for local testing

- A. Requirements:
  - a. Install Docker
    - Purpose: In case there is no Raspberry pi available you can test it via docker as follows:
  - b. Clone repo <https://github.com/sprintcube/docker-compose-lamp>
- B. Copy and unzip docker-compose-lamp.zip into your repo ( .gitignore)

### 3. Follow Readme:

a) cd docker-compose-lamp

b) docker compose up

Executing it for the first time will take some time

c) open lamp

### 4. Execute database scheme

### 5. Run app.py

### Troubleshooting:

1. Is Docker Running?
2. Are ports already in use?
3. Set user and password with rights.

### 3. What is part of the application

#### A. App.py

Contains all routes used in the application and is responsible for running the app. The routes contained in this file are the following:

- `/`: route of the main page of the application
- `/offers`: route of the offers page, where all offers are displayed
- `/events`: route of the events page, where all events are displayed. This route also contains both GET and POST methods

#### B. Data.py

Responsible for the database calls. Contains all queries used in the application.

#### C. Sqlbuilder.py

Responsible for the database connection and running the queries.

#### D. Database.sql

Contains the database creation query, as well as the content of the database, such as the tables.

The application also contains a static folder, which includes the HTML templates used in the app as well as the CSS styles.