Project Save Baltimore

-

Administration Manual (AM)

Version 1.0

Produced For:
Next Century Corporation

Produced By:
Team Indigo (CMSC447):

- Neil Joshi
- Nat Baylon
- Matthew Landon
- Bernie McNamee

Date: May 5, 2016

Save Baltimore
Administrator Manual

Table of Contents

1. Introduction

1.1 Purpose of This Document

This document will contain all of the required information needed for adopting the Baltimore crime visualiser. It is intended to be read by the future system administrator of this system. The following sections will first explain what is required to run the application. Following this is the process needed to deploy this system for the first time. Next will be tasks tasks which will need to be performed during the lifecycle of the system. Finally, this details known issues of the system and how these should be handled to maintain a productive system.

1.2 References

Systems Requirements Specification (SRS) by Team Indigo

System Design Document (SDD) by Team Indigo

2. System Overview

2.1 Background

The system administrator will need to monitor working functionality of the AWS servers. This may mean to increase memory and CPU processing speed, if throughput and latency becomes too slow -this also increases monthly cost for hosting the application (something to consider). Knowledge of the directory structure is also important: the system has two subdirectories one containing all database files and the other with all web files. All web apis should go under the CMSC447_proj/libs/ folder. The administrator should also acknowledge that there is a monthly data retrieval (via python script from the Baltimore crime website, into the DB), and should make sure that process continues to run. Updating and installing new functions and libraries to php5 can be done in users/bin/php5. Currently, backups of all program files can be found on GitHub.

2.2 Hardware and Software Requirements

Hardware and Software dependencies/requirements are as follows: The system needs a web hosting server and a connecting DB server for data queries. Likewise mySQL will need to be installed and set up on the server. Php 5 will need to be configured correctly, by enabling the mysqli lib. The SlickGrid Javascript library will also need to be downloaded and placed within the project folder. To use the Google heat map api, a valid project number must be registered with the Google developers site. Finally a web browser, running HTML5 and JS will be needed.

3. **Administrative Procedures**

3.1 Installation

Starting from a bare computer/server, the first step will be to install the required supplementary applications. The first is an apache2 web server. This will set up a hosting environment for the website.
Php needs to be installed from the default updater. Enable mySqli in the PHP.ini file.

Also, install the software packager: gnu wget. This will be used to download the csv file from the open baltimore data source.

Finally, mySQL server should be installed. During installation, for the root user, add the password “cmsc447”.

Copy the DBFiles folder into /home/<user>
From the DBFiles directory, run the following commands:

Create the database with a password:
$mysql -u root -p
<enter the password: cmsc447>
mysql> create database save_baltimore;

Then, set up the cron job to autoupdate the dataset. In the /etc/cron.monthly directory, add the file UpdateSaveBaltimore.sh found in DBFiles. In the second line of UpdateSaveBaltimore.sh, replace “password” with the password of the user of the linux system, and make sure that user has the privileges to remove from /home/<user>/DBFiles

Finally, run the bash script: UpdateSaveBaltimore.sh to initially populate the database.

Copy the source code folder: CMSC_447_proj into the hosting folder, probably in /var/www to set up the website.

3.2 Routine Tasks

This site is autonomous. The data is automatically updated via the cronjob. As such, no routine tasks are required for successful usage of the application.

3.3 Periodic Administration

Once the site is deployed, the application does not need to have interference because it is self contained and manages the outside data dependencies (OpenBaltimore).

Discuss any tasks to be performed periodically such as system backups and the cleaning up of user accounts.

3.4 User Support

The development team will be stepping away from this system and is unable to provide further support. However, the associated documentation should provide the necessary support for any questions that may arise.

4. **Troubleshooting**

4.1 Dealing with Error Messages and Failures

If data is not appearing on the site, the dependent services need to be inspected. First, look at the data provider to check for errors. Make sure the project key-registered with google (for the heat map)- is still valid. A new key may be needed. Other than this, the site should not be creating errors.

4.2 Known Bugs and Limitations

If the server has not been provided with the correct CPU processing and memory, the application will be very slow and detrimental to the user's experience. The only solution here is to upgrade the server or decrease the number of records that are pulled from the database in data.php

Appendix A – Team Review Sign-off

Nat Baylon	5/5
Matthew Landen	5/5
Neil Joshi	5/5
Bernie McNamee	5/5

Appendix C – Document Contributions

Identify how each member contributed to the creation of this document. Include what sections each member worked on and an estimate of the percentage of work they contributed. Remember that each team member must contribute to the writing (includes diagrams) for each document produced.

Nat: 3.1
Matthew: 1, 3.2,3.3, 4.1,
Neil: 4.2 3.1
Bernie: 2.1,2.2