System Documentation / Programmer's Manual

parkingFriendly

December 4, 2018

Team Members

Nicholas Greer

Michael Mogannam

Gabriel Cabrera

Contents

Revision History

Version	Date	Name	Description
1	12/02/18	Gabriel Cabrera, Michael Mogannam, Nicholas Greer	Initial Document
2	12/04/18	Gabriel Cabrera, Michael Mogannam, Nicholas Greer	Updated document

1. Introduction

The parkingFriendly application allows users a simple way to manage their parkingFriendly Credit account information from a web browser. This document will provide instructions for accessing the application code and setting up the database for use to allow use of the web app.

2. Setup Database

A Unix/Linux based server with internet connectivity is required. Access to some sort of SSH/FTP protocol is also required to be able to access the server in which the files are stored:

nrs-projects.humboldt.edu

Obtain the proper Username and Password to ssh into server.

1. 1.1.1 parkingFriendly@nrs-projects.humboldt.edu port:22
   1. 1.1.1. SSH into this account for access to server.
2. Loging into SQLPLUS
   1. 2.1.Navigate to appropriate directory(cs458)
      1. 2.1.1.Write out the command “sqlplus” + ENTER
         1. 2.1.1.1. Log in using the same credentials
            1. 2.1.1.1.1 run “create-table.sql” file to create tables for DB

3. Access Application Code

Internet Access is required to be able to view all Application Code. It may be accessible via ssh into the user account listed above or can be accessed via the web.

All code can be accessed via link to this github account:

https://github.com/gjc129/CS458-Project

4. System Maintenance

1. *HTML Code*

The user interface consists of four tabs for viewing information, plus the login screen. Each of these is implemented in one php file. There are a few additional php files: the application parkingFriendly view ("home.php"), tab bar navigation view ("parkingFriendlyCreditTabs.php") and registration page(“registration.php”).

On the first run, the application displays the login screen. Once the user has logged in, the view is switched to the tab bar, which is displayed at the bottom of the screen. The tab bar view enables the user to navigate between the tabs mentioned above, and hosts the four subviews for each type of information.

The logic in each view controller class mainly performs the task of pulling the appropriate data from XMLParser, formatting it, and displaying it in the correct fields. The login controller class also contains logic to store the user's credentials in the system Keychain.

2. *PHP Code*

The proxy server consists of php, executed by apache when a specific POST is received via https. The php code logs into nrs-projects using the user’s credentials (provided in POST). From there, the code captures an https re-direct, and uses the unique URL in the re-direct to request account balance and account history. This is accomplished in three separate https transactions.

As information is gleaned from the second two transactions, the php script generates oraclesql, and sends it to the db on the device through the (still open) tcp stream.

When the parsing and information generation is complete, the tcp session is closed, and the php script exits. No passwords or user data is retained on the proxy server.