# WEB DATA PROCESSOR & SMALL SEVER DEVELOPMENT

**Sebastian Gonzalez**

**Mailee Chue**

**ITSE 1359 - Introduction to Scripting**
**El Paso Community College**

# Table of Contents

## I . PROJECT DESCRIPTION

This project consists of creating a small web server with a database capability using a raspberry pi. This particular project has the purpose of helping a local company to launch a website online. The use of the programming languages: Python and SQL will be the tools used to develop the web server. The web server will include secure scripting techniques that will protect all of the information belonging to the system and information stored in the databases.

The system will be developed under the mentorship of Christian Servin, PhD. ACM/IEEE guidelines will be used to effectively create the software.

## II. ROLES AND DUTIES

These responsibilities were well adjustable throughout the members on certain part of the project. The four roles and duties within the project which were:

- Lead programmer
- Report writer
- Time manager and deadline vertifier
- Integration and validation

Sebastian Gonzalez took the responsibility as the lead programmer and report writer when it comes to developing the web server, and database scripting(MySQL). The web server will be built under the cybersecurity frameworks from NIST, IEEE, and ACM. The software will be developed in a Raspberry Pi 3 computer.

Mailee Chue took the responsibility as the time manager and deadline verifier. Moreover, she also took the lead to developed a constructional website called "Constructora Rios Maciel." Created the website in Adobe Dreamweaver and Photoshop CS6. The site development consisted of basic HTML and CSS. The company has its own logo, so there was no need to customize a new logo, but only need to be redesign into vectors, show **figure 1.1** and **1.2.**

## III. CLIENT

**About Constructora Rios Maciel**

Constructora Rios Maciel is a private family-owned construction business company founded by Jesus Rios Maciel, located in Ciudad Juarez, Chihuahua, Mexico. The company is 17 years old and has a staff of only 3 persons. Constructora Rios Maciel performs all types of construction work in the city, from small projects such as pool construction and rubble cleanups to big projects such as tunnel construction, highway maintenance, and demolitions. Some of the companies Constructora Rios Maciel has worked for:

- **Ferromex**: Ferromex is a private rail consortium that operates the largest railway in Mexico.
- **Federal de Electricidad:** The Comisión Federal de Electricidad is the state-owned electric utility of Mexico, widely known as CFE. It is the country's dominant electric company.

## IV. PURPOSE

The purpose of creating the company's website and web-server is to expand their services throughout the city, state and nation. Constructora Rios Maciel has only been advertised in the local newspaper and social media. Due to economy changes, Constructora Rios Maciel has to reach out to the online community and enter a brand new form of online advertisement. This project represents a breakthrough in Constructora Rios y Maciel long history. Director, Jesus Rios Maciel, commented "I always wanted to publish a website online, but lacked the tools and knowledge to do so. This will change how people can reach us, and we can reach them." Local businesses have to rely in constant paid advertisement, which only remains local.

Having the benefit of a small low-cost web server expands the possibility of small businesses to become largely known and recognized. As stated by Karen Mills, a publicist and reporter from fortune.com, small businesses have a better chance of growing when using online services and tools.

Another benefits that this project will bring, is that students involved will learn how to create websites, using Adobe, databases using MySQL, and web servers. Other skills that students in this project will learn are, time management, project management, validation and verification, database management, web development, secure scripting, and teamwork.

## V. RELATED PROJECTS

Other Raspberry Pi web servers have been used to integrate a music and video service, web video surveillance for personal items, pets, or home and business, and to make intelligent systems for home. Some examples are, automatization of tasks, home security, lighting control, and door control.

## VI. GUIDELINES & LEARNING OUTCOMES

**From NIST:**

1. Maintain database management systems software.
2. Monitor and maintain databases to ensure optimal performance
3. Perform backup and recovery of databases to ensure data integrity
4. Apply secure code documentation.
5. Assess the effectiveness of cybersecurity measures utilized by system(s).
6. Collect and maintain data needed to meet system cybersecurity reporting.
7. Characterize and analyze network traffic to identify anomalous activity and potential threats to network resources.
8. Monitor and report client-level computer system performance.
9. Knowledge of client organizations, including information needs, objectives, structure, capabilities, etc.

**From CCECC:**

1. Investigate vulnerabilities and failure scenarios in database systems, such as SQL injection and cross-site scripting.
2. Analyze security vulnerabilities in various data structures
3. Recognize the importance of security as a continuous process of tradeoffs.
4. Differentiate the concepts of risk, threats, vulnerabilities, attack vectors, and exploits.
5. Choose among various countermeasures and security controls to minimize risk and exposure in a given scenario.
6. Use the principles of secure design, such as least privilege, isolation, fail-safe, and deny-by-default.
7. Identify the components of a database system.
8. Formulate queries in SQL or a similar query language to elicit information from a database.
9. Apply security principles and practices in a dynamic environment.

**From CSEC:**

1. The system will maintain confidentiality. Rules that limit access to system data and information to authorized persons.

2. The system will maintain integrity. Assurance that the data and information are accurate and trustworthy.
3. The system will have availability. The data, information, and system are accessible.
4. Risk analysis. Potential for gain or loss.

## VII. DEVELOPMENT PLAN

- **Week 1:**

Time frame of developing the website's structure. The first week is to be used to create the skeleton of the site. Hand sketching the design of the website.

- **Week 2:**

Storing information onto the website. Since the construction company is located in Mexico, the site will only be written in spanish. The currency will be in pesos. However, for the presentation of this project, the information will be presented in english. The product delivered will be in spanish.

- **Week 3:**

Develop the web-server using Django software. The HTML and related files will be prepared for insertion into the Django system. Beforehand, the Django server will be setup in a Raspberry Pi 3 Model B.

- **Week 4:**

Test services and troubleshoot errors.
Document files. Revise integrity, security, flaws of the system. Gather with client to check all requirements. Work on new requests by client, if any.

- **Week 5:**

Delivery of product to the client. Maintenance state begun.

**Designing Aspect**


**Figure 1.1:** Before Logo Pixel

Constructora Rios Maciel logo had been reconstructed in Adobe Illustrator CS6 for a more clean finish, show in Figure 2.


**Figure 1.2:** After Logo Vector

**Services to user from website**

**Quote Survey:**
The created survey for the user is only to help with getting a quote for rental equipment or the fix rate for constructing a pool. The user will be prompted to enter their email, and fullname in order to receive the direct quote from employees at Constructora Rios Maciel. Note: requesting for a quote will not store email address from users in the database.

**Newsletter Service**

Users will be able to subscribe to the website to receive updates on projects or general announcements. To subscribe, users will have to create a free account and enter their full name

and email address. User information will be saved and protected in a database.

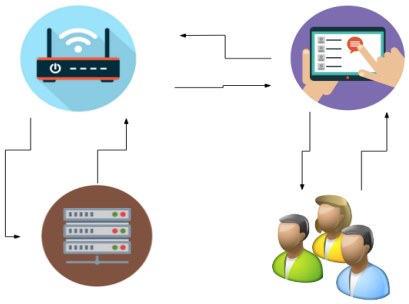

**Figure 2**

## VIII. ENTITY RELATIONAL DATABASE

Overall there are two entities and 7 attributes to develop a quote for users who wants the services from Rios Maciel Constructora. The table holds the name of each entity, list of attributes, type of data the fields are accepting, and a description of the duties of each entity, show in Table 5.

| Entity | Attribute Name | Type | Entity Description |
|---|---|---|---|
| **Services_T** | **(PK)** ServiceID | INTEGER | This entity is for any kind of charges that is going to be made to the user such as, late fee, normal fee, damage fee, or lost fee. |
| | Service_Name | VARCHAR2(10) | |
| | Service_Price | NUMBER(4,2) | |
| **User_T** | **(PK)** UserID | INTEGER | This entity stores each user full name and email. |
| | User_Fname | VARCHAR2(25) | |
| | User_Lname | VARCHAR2(25) | |
| | User_Email | NVARCHAR(30) | |

**Table 5: List of Entities, Attributes, Types, and Description**

**CREATE DATABASE**

**Scripting SQL File:** Setting up the CreateDatabase file. A spool file "CreateDatabase.log" will display the input code in **">SQL"** format showing that it has been run and all the performance the program has to do to create the table. By setting the pagesize and linesize will prevent data from overlapping on display.

```
SPOOL CreateDatabase.log;

SET PAGESIZE 100
SET LINESIZE 180
SET SERVEROUTPUT ON
SET ECHO ON
```

**Scripting SQL File:** Drop Table usually show that every table has been executed successfully. If not, there will be an error and a statement to show where the error is coming from.

```
DROP TABLE Service_T CASCADE CONSTRAINT;
DROP TABLE User_T CASCADE CONSTRAINT;
COMMIT;
```

**Scripting SQL File:** The following tables created are two tables in total and nine attributes,two primary keys. Once all tables are created then the file close.

```
CREATE TABLE Services_T(
ServiceID          INTEGER AUTOINCREMENT,
Service_Name       VARCHAR2(10),
Service_Price      NUMBER(4,2)
CONSTRAINT Service_PK PRIMARY KEY(serviceID));

CREATE TABLE User_T(
          UserID           INTEGER AUTOINCREMENT,
          UserName         VARCHAR2(25),
          UserLast VARCHAR2(25),
          UserEmail        NVARCHAR(30),
CONSTRAINT User_PK PRIMARY KEY(userID));
```

```
SET SERVEROUTPUT OFF
SPOOL OFF
```

---

**INSERT INTO DATABASE**

**Scripting SQL File:** Setting up the InputData file. A spool file "InputData.log" will display the table. The pagesize and linesize is to prevent the columns in the table to overlap if the size of the table has reached its max. By setting a certain size will prevent this from happening, to create a complete table display.

---

```
SPOOL InputData.log;

SET PAGESIZE 95
SET LINESIZE 180
SET SERVEROUTPUT ON
SET ECHO ON
```

---

**Scripting SQL File:** If the table has anything in them then it needs to be empty.

---

```
DELETE FROM Charge_T CASCADE CONSTRAINT;
DELETE FROM User_T CASCADE CONSTRAINT;
```

---

**Scripting SQL File:** Insert information into each table then close the file once all insertion has been input. Turn Off ServerOutPut and Spool.

---

```
INSERT INTO User_T (User_Fname,
User_Lname,User_email)
VALUES ('John','Doe','johnDoe@email.com');

INSERT INTO Services_T (Service_Name,Service_Price)
VALUES ( 'Dump Truck Rent', 50);

SET SERVEROUTPUT OFF
SPOOL OFF
```

---

**FINAL DATABASE**

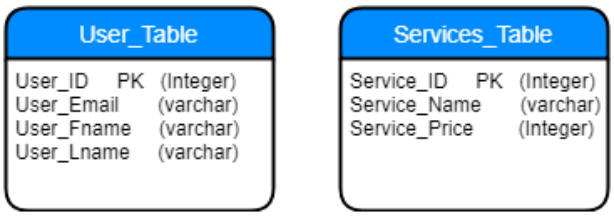


**Figure 3:** Rios Maciel Constructora Database

***Database Description:***
The final database is only use for the company to keep track of their customers and the kind of services being demand by each individual customer.

## VIX. WEB SERVER DEPLOYMENT

The following will describe the tools and services used for the development of the web server and HTML processing.

This project was developed using a system developed in python scripts named Django.
This software allows developers to handle the backend production as well as the frontend. Django follows a layered structure divided in modules or folders.

The Django web-server consists primarily of two layers. In the first layer, Django contains the database file that will handle all database related actions. For example, creating users, managing tables, and encryption.

Django comes with the ability of hashing stored passwords before inserting them into the database. Giving the developer less to worry about.

Another file present in the first layer of Django, is the manage.py script. This script can perform various operations such as starting the server, creating a super-user, migrating

changes to database, creating new applications, and more.

In this first layer, the developer must insert new folders that will contain the files to be inserted into the server. For example, the developer must create a new folder called "templates", and in this folder the developer must insert all the HTML files related to the project.

When the command "python manage.py startapp [app name] " is run, the script manage.py will create a new folder which will contain the all the settings-related scripts that the application will contain. This is the second layer of Django, the application and files.

The second layer contains the files _init_.py, views.py, settings.py, urls.py, wsgi.py. All of these files are in charge of obtaining all the information from other folders so that the server can display the HTML and run the database. **Figure 4** shows the structure of the Django web-server.

Constructora Rios Maciel's configuration settings scripts contain the following:

## settings.py

In here the developer can edit the authorized users, define the localhosts, set static files, and other configurations. Code sample:

```
import os

# Build paths inside the project like this:
os.path.join(BASE_DIR, ...)
BASE_DIR =
os.path.dirname(os.path.dirname(os.path.abspath(__file
__)))

# Quick-start development settings - unsuitable for
production
# See
https://docs.djangoproject.com/en/1.11/howto/deploy
ment/checklist/

# SECURITY WARNING: keep the secret key used in
production secret!
SECRET_KEY =
not shown for security purposes

# SECURITY WARNING: don't run with debug turned on in
production!
DEBUG = True

ALLOWED_HOSTS = ["localhost","0.0.0.0"]

# Application definition

INSTALLED_APPS = [
   'django.contrib.admin',
   'django.contrib.auth',
   'django.contrib.contenttypes',
   'django.contrib.sessions',
   'django.contrib.messages',
   'django.contrib.staticfiles',
]
MIDDLEWARE = [
   'django.middleware.security.SecurityMiddleware',

'django.contrib.sessions.middleware.SessionMiddleware'
,
   'django.middleware.common.CommonMiddleware',
   'django.middleware.csrf.CsrfViewMiddleware',

'django.contrib.auth.middleware.AuthenticationMiddlew
are',

'django.contrib.messages.middleware.MessageMiddlewa
re',

'django.middleware.clickjacking.XFrameOptionsMiddlew
are',
]
ROOT_URLCONF = 'raspberrypiapp.urls'

TEMPLATES = [
   {
      'BACKEND':
'django.template.backends.django.DjangoTemplates',
      'DIRS': ['templates'],
      'APP_DIRS': True,
      'OPTIONS': {
         'context_processors': [
            'django.template.context_processors.debug',
            'django.template.context_processors.request',
            'django.contrib.auth.context_processors.auth',
```

```
'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'raspberrypiapp.wsgi.application'


# Database
#
https://docs.djangoproject.com/en/1.11/ref/settings/#d
atabases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}


# Password validation
https://docs.djangoproject.com/en/1.11/ref/settings/#a
uth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeS
imilarityValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.MinimumLengt
hValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.CommonPassw
ordValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.NumericPassw
ordValidator',
    },
]

# Internationalization
# https://docs.djangoproject.com/en/1.11/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True

# Static files (CSS, JavaScript, Images)
#
https://docs.djangoproject.com/en/1.11/howto/static-fil
es/

STATIC_URL = '/static/'
STATICFILES_DIRS = (
    os.path.join(BASE_DIR,'assets/images'),
    os.path.join(BASE_DIR,'assets/css'),
    os.path.join(BASE_DIR,'templates'),
)
```

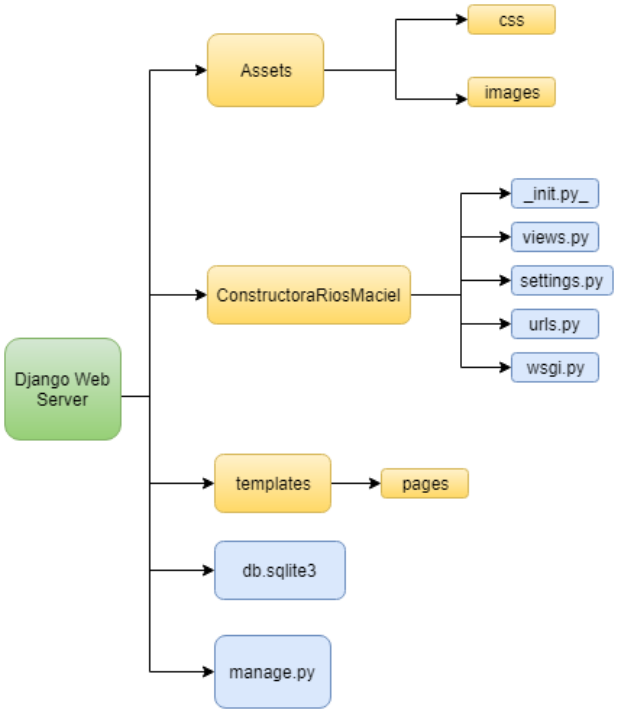


**Figure 4:** Django web-server treeview

## X. FINAL WEBSITE

***Keep it Simple :***

Sometimes plans may not always work out as planned. Sometimes ideas on paper may be harder to put together. The issues which I encountered as I tried to maintain with a simple design to not delay the results the project. However, the plan did not go as planned. The chosen color above was harder to work with than I thought. It was easy to come up with a simple design, but it was hard to really be happy with it. Therefore, I had to make some changes with my color scheme. I relooked at the logo and select other colors to work with rather than staying with the color scheme I had before. The colors which were chosen were a brownish red, lavender, and grey. Just with simple changes I was able to come up with a more today look website for the construction business.

***Is Simple Good Enough? :***

From the beginning, I was going to design the website in Photoshop, use the slice tool, and transfer the layout to Dreamweaver. However, I ran into some problems with presenatable quality work. Therefore, instead of taking the simple route of using HTML and little of CSS, I then have to incorporate more to the CSS instead of just using it for the navigation bar and background for each individual divs.


**Figure 5:** ThumbNail Design

## REFERENCES


1. Karen Mills. 2014. How the Internet is giving Small Business Saturday an edge. (November 2014). Retrieved July 9, 2018 from http://fortune.com/2014/11/29/how-the-internet-is-giving-small-business-saturday-an-edge/

2. Nuttall, B. (2017, April 14). 5 projects for Raspberry Pi at home. Retrieved July 9, 2018, from https://opensource.com/article/17/4/5-projects-raspberry-pi-home

3. "Documentation," Security in Django | *Django documentation* | Django. [Online]. Available: https://docs.djangoproject.com/en/2.0/ref/settings/. [Accessed: 02-Aug-2018].

4. W. Newhouse, S. Keith, B. Scribner, and G. Witte, *National Initiative for Cybersecurity Education (NICE) Cybersecurity Workforce Framework*. National Institute of Standards and Technology , 2017.

5. ACM, IEEE, AIS, IFIP, *Curriculum Guidelines for Post-Secondary Degree Programs in Cybersecurity*. Association for Computing Machinery, 2017.

6. E. K. Hawthorne, C. Tang, C. S. Tucker, C. Servin, and T. Moore, *Computer Science Curricular Guidance for Associate-Degree Transfer Programs with Infused Cybersecurity*. Committee for Computing Education in Community Colleges, 2017.