

ReUse Website Notes

Website File Structure

App

- **Api**
 - **AdminRoutes.php – Routes used by the admin portal to change or view the database.**
 - GET '/admin' – Redirect to /AdminSite2/loginPage.php
 - GET '/RUapi' group:
 - GET '/category/:id' – Returns array of search results from DB using query 'SELECT name, id FROM Reuse_Categories WHERE Reuse_Categories.id = '.\$id.'" Returns data with specific ID.
 - GET '/states' – Returns array of search results from DB using query 'SELECT name, id FROM States'. Returns all states.
 - GET '/business' – Returns array of search results from DB using query 'SELECT name, id, address_line_1, address_line_2, state_id, phone, website, city, zip_code FROM Reuse_Locations'. Returns all businesses.
 - GET '/category' – Returns array of search results from DB using query 'SELECT name, id FROM Reuse_Categories'. Returns all categories.
 - GET '/items/:id' – Returns array of search results from DB using query 'SELECT name, id, category_id FROM Reuse_Items WHERE Reuse_Items.id = '.\$id.'" Returns items with a specific ID.
 - GET '/business/:one' – Returns array of search results from DB using query "SELECT name, id, address_line_1, address_line_2, state_id, phone, website, city, zip_code FROM Reuse_Locations WHERE Reuse_Locations.id = '\$one'". Same as a function for ID, but with 'one' for variable.
 - GET '/items' – Returns array of search results from DB using query 'SELECT name, id, category_id FROM Reuse_Items'. Returns all items.
 - GET '/businessdocs/:id' – Returns array of search results from DB using query "SELECT id, name, URI FROM Reuse_Documents WHERE location_id = '\$id'". Returns a doc for a specific ID.
 - DELETE '/RUapi' group:
 - DELETE '/business/:id' – Updates the database with deleting a specific business using ID. Utilizes the query, 'DELETE FROM Reuse_Locations_Items WHERE location_id = '\$delID'".
 - DELETE '/item/:id' – Updates the database with deleting a specific item using the ID. Utilizes the query, 'DELETE FROM Reuse_Locations_Items WHERE item_id = '\$delID'".
 - DELETE '/category/:id' – Updates the database with deleting a specific category using the ID. Utilizes the query, 'DELETE FROM Reuse_Categories WHERE Reuse_Categories.id = '\$delID'".

- DELETE '/updateBusiness/:one/:id' – Updates the database with deleting a specific item from a business's catalog. It first gets a list of ID and names of the business from the location, matches names with the one parameter, removes where there is a match.
- DELETE '/businessDoc/:id' – Removes a specific document from a business. Utilizes the query, 'DELETE FROM Reuse_Documents WHERE id = '\$delID''
- PUTS '/RUapi':
 - POST '/updateDataForMobile' – Calls reuse_generateXML() to update reuse_database.xml
 - POST '/changeCategory' – Updates the name of a category using the following query, 'UPDATE Reuse_Categories SET name = '\$name' WHERE name = '\$oldName''
 - POST '/changeItem' – Updates the name of an item using the following query either by name, 'UPDATE Reuse_Items SET name = '\$name' WHERE name = '\$oldName'', OR with category, 'UPDATE Reuse_Items SET category_id = '\$cat' WHERE name = '\$oldName''
 - POST '/changeBusiness' – Makes changes to the business's attributes (name, zip, state, etc.). Sample query is as follows, 'UPDATE Reuse_Locations SET phone = '\$phone' WHERE name = '\$oldName''
- POSTS '/RUapi':
 - POST '/business' – Insert a Business with all it's attributes. Query is as follows, 'INSERT INTO Reuse_Locations (name, address_line_1, address_line_2, city, state_id, zip_code, phone, website, latitude, longitude) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)'
 - POST '/category' – Insert a category. Query is as follows, 'INSERT INTO Reuse_Categories (name) VALUES (?)'
 - POST '/updateItems' – Update category item by the item name. Query as follows, 'UPDATE Reuse_Items SET category_id = '\$category' WHERE Reuse_Items.name = '\$name''
 - POST '/updateBusiness' – Update the relationship table between items and location. Query is as follows, 'INSERT INTO Reuse_Locations_Items (item_id, location_id) VALUES (?, ?)'.
 - POST '/items' – Insert an item using the following query, 'INSERT INTO Reuse_Items (name, category_id) VALUES (?, ?)'
 - POST '/addBusinessDoc' – Insert a business doc to the database using the following the query, 'INSERT INTO Reuse_Documents (name, URI, location_id) VALUES (?, ?, ?)'
- **LatLongRoutes.php – Route which updates businesses with latitude and longitude for geocoding.**
 - GET '/setLatLongs' – Goes into the database and selects locations where the attributes for latitude and longitude are set to NULL. With each location under this condition, it uses the `bingGeocode(address, city, state, zipcode)` function to get latitude and longitude. Finally updates the database with the location's

latitude and longitude with the following query, 'UPDATE Reuse_Locations SET latitude = '\$latitude', longitude = '\$longitude' WHERE id = '\$id'

- **PhoneRoutes.php**

- GET '/reuseDB' – Echo or print XML tree file of the contents of the database. Specifically, a list of businesses and data relevant to the businesses.
- GET '/recycleNameXML' – Echo an XML tree with a list of names of recycling centers ordered by name.
- GET '/recycleXML' – Echo an XML tree with a list of recycling centers and data relevant to the businesses.
- GET '/donorXML' – Echo an XML tree of the donors/sponsors and their attributes.

- **WebsiteRoutes.php**

- GET '/' – Redirects to '/HomeSite/home.php'
- GET '/business/reuseExclusive' – Returns an array of businesses not in Repair, Repair Items, or Recycle and not having the recycle flag set to 1.
- GET '/business/recycleExclusive' – Returns an array of businesses that are recycling centers only and having the recycle flag set to 1.
- GET '/business/category/name/:cat_name' – Returns a businesses in order by name under a specific category.
- GET '/business/category/name/not/:cat_name' – Returns businesses in order by name NOT under a specific category.
- GET '/business' – Return all businesses in the database along with their attributes.
- GET '/business/item/name/:item_name' – Returns businesses in order by name with a specific item.
- GET '/business/category/name/:cat_name/item/name/:item_name' – Returns businesses in order by name within a category and has a specific item.
- GET '/business/name/:bus_name' – Returns the first instance with the given name.
- GET '/item/business/name/:bus_name' – Returns an array of items with a given business name.
- GET '/item/category/name/:cat_name' – Returns an array of items with a given category.
- GET '/item/category/reuseExclusive' – Returns distinct items from a specific category.
- GET '/category/reuseExclusive' – Return distinct category names not including Repair, Repair Items, and Recycle.
- GET '/document/business/name/:bus_name' – Return an array of documents/links associated with a given business.
- GET '/donor' – Return an array of distinct donors.

- **Database**

- **database_cred.php**
- **reuseConnect.php**

- **Lib**

- BingGeocoder
 - geocoder.php
 - myxml.xml
 - testGeoCoder.php
- xmlGenerator
 - reuse_database.xml
 - xmlGenerator.php
 - xmlGeneratorConfig.php

public_html

- AdminSite
 - **addBusiness.php – Form to add a new business with business name, address, city, state, zip code, phone number, and website.**
 - Css – bootstrap.css, customStylesheet.css, media.css, //maxcdn.bootstrapcdn.com/font-awesome/4.3.0/css/font-awesome.min.css, 'https://fonts.googleapis.com/css?family=Rubik:700' rel='stylesheet' type='text/css'.
 - js - jquery-1.11.1.min.js, BusFunct.js, bootstrap.min.js
 - **addCategory.php – Form to add a new category with adding only name of category.**
 - Css - bootstrap.css, customStylesheet.css, media.css, //maxcdn.bootstrapcdn.com/font-awesome/4.3.0/css/font-awesome.min.css, 'https://fonts.googleapis.com/css?family=Rubik:700' rel='stylesheet' type='text/css'.
 - js - jquery-1.11.1.min.js, jquery.multi-select.js, CatFunct.js, bootstrap.min.js
 - **addItem.php – Form to add a new item with item name.**
 - Css - bootstrap.css, customStylesheet.css, media.css, //maxcdn.bootstrapcdn.com/font-awesome/4.3.0/css/font-awesome.min.css, 'https://fonts.googleapis.com/css?family=Rubik:700' rel='stylesheet' type='text/css'.
 - js - jquery-1.11.1.min.js, ItemFunct.js, bootstrap.min.js
 - **categoriesToAdd.php – Lists categories**
 - Css - bootstrap.css, customStylesheet.css, media.css, //maxcdn.bootstrapcdn.com/font-awesome/4.3.0/css/font-awesome.min.css, 'https://fonts.googleapis.com/css?family=Rubik:700' rel='stylesheet' type='text/css'.
 - js - jquery-1.11.1.min.js, BusFunct.js, bootstrap.min.js
 - **loginCheck.php – To securely login a user into the system as well as registering a new user and ending a user's session. All PHP.**
 - **loginPage.php – Form with username, password, and submit button to login.**
 - Css - bootstrap.css, customStylesheet.css, media.css, //maxcdn.bootstrapcdn.com/font-awesome/4.3.0/css/font-awesome.min.css, 'https://fonts.googleapis.com/css?family=Rubik:700' rel='stylesheet' type='text/css'.

- js – loginFunc.js, //ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js, bootstrap.min.js
- **logout.php – Asks if user wants to logout, if user selects button to logout the logOut() function is called.**
 - Css - bootstrap.css, customStylesheet.css, media.css, //maxcdn.bootstrapcdn.com/font-awesome/4.3.0/css/font-awesome.min.css, 'https://fonts.googleapis.com/css?family=Rubik:700' rel='stylesheet' type='text/css'.
 - js - jquery-1.11.1.min.js, bootstrap.min.js
- **main.php – Main page to make Database Management Options for Business, Category and Item.**
 - Css - bootstrap.css, customStylesheet.css, media.css, //maxcdn.bootstrapcdn.com/font-awesome/4.3.0/css/font-awesome.min.css, 'https://fonts.googleapis.com/css?family=Rubik:700' rel='stylesheet' type='text/css'.
 - js - jquery-1.11.1.min.js, mainFunc.js, bootstrap.min.js
- **nav.php – A navigation bar on the left of display to list Admin functions (i.e. Add/Edit/Delete Business).**
- **oneTimeRegister.php – A form which executes a newUser(), gets username and password. Checks if either are null, otherwise sends XMLHttpRequest.**
 - Css - bootstrap.css, customStylesheet.css, media.css, //maxcdn.bootstrapcdn.com/font-awesome/4.3.0/css/font-awesome.min.css, 'https://fonts.googleapis.com/css?family=Rubik:700' rel='stylesheet' type='text/css'.
 - js - jquery-1.11.1.min.js, bootstrap.min.js
- **Register.php – Creates an account with a username and password and sends it to be processed with the newUser(). newUser(), gets username and password, checks if either are null otherwise sends an XMLHttpRequest to DB.**
 - Css - bootstrap.css, customStylesheet.css, media.css, //maxcdn.bootstrapcdn.com/font-awesome/4.3.0/css/font-awesome.min.css, 'https://fonts.googleapis.com/css?family=Rubik:700' rel='stylesheet' type='text/css'.
 - js - jquery-1.11.1.min.js, bootstrap.min.js
- **searchBusiness.php – Form to Edit/Remove a listed business with the business's attributes, or search for a business. The searchBusFunc script has searchBusiness() to find a business, and also has allBusiness() to list all businesses.**
 - Css - bootstrap.css, customStylesheet.css, media.css, //maxcdn.bootstrapcdn.com/font-awesome/4.3.0/css/font-awesome.min.css, 'https://fonts.googleapis.com/css?family=Rubik:700' rel='stylesheet' type='text/css'.
 - js – searchBusFunc.js
- **searchCategory.php – Form to Edit/Remove a listed categories with the category name, or search for a category. The searchCatFunc script has**

searchCategoryByName() to find a category, and also has **allCategories()** to list all businesses.

- Css - bootstrap.css, customStylesheet.css, media.css,
//maxcdn.bootstrapcdn.com/font-awesome/4.3.0/css/font-awesome.min.css,
'https://fonts.googleapis.com/css?family=Rubik:700' rel='stylesheet'
type='text/css'.

- js - jquery-1.11.1.min.js, bootstrap.min.js, searchCatFunc.js

- **searchItem.php – Form to Edit/Remove a listed categories with the category name, or search for a category. The searchCatFunc script has searchCategoryByName() to find a category, and also has allCategories() to list all businesses.**

- Css - bootstrap.css, customStylesheet.css, media.css,
//maxcdn.bootstrapcdn.com/font-awesome/4.3.0/css/font-awesome.min.css,
'https://fonts.googleapis.com/css?family=Rubik:700' rel='stylesheet'
type='text/css'.

- js - jquery-1.11.1.min.js, bootstrap.min.js, searchItemFunc.js

- **AdminSite2**

- **components**

- **css-comp-css**

- centeredNaviDraggable.css – centers stuff to the middle of the page.
 - gridComponent.css – grid styling
 - largeDraggable.css - styling
 - naviDraggable.css – styling.

- **img-comp-img**

- **foundation-icons**

- **svgs – many things here for svgs.**
 - .fontcustom-data
 - foundation-icons.css
 - foundation-icons.eot
 - foundation-icons.svg
 - foundation-icons.ttf
 - foundation-icons.woff
 - preview.html

- **js-comp-js – javascript to help make things draggable.**

- centeredNaviDraggable.js
 - largeDraggable.js
 - naviDraggable.js

- **loginStuff**

- checkSession.php – check the session state has a username.
 - destroySession.php – Kill the session when so desired.
 - loginCheck.php – Securely logging a user in to the system, registering a new user and ending a user's session.

- centeredNaviDraggable.php

- gridComponent.php

- largeDraggable.php

- naviDraggable.php
- css – styling for each page
 - addBusinessPage.css
 - addCategoryPage.css
 - addItemPage.css
 - allBusinessesPage.css
 - allCategoriesPage.css
 - allItemsPage.css
 - loginPage.css
 - main.css
 - singleBusinessPage.css
 - singleCategoryPage.css
 - singleItemPage.css
- img
 - sky-background.jpg – image of the sky
 - treebark.png – image of artistic looking treebark.
- js
 - addBusinessPage.js – Appends HTML to add list item to list in allBusinessesPage.php.
 - Uses AJAX to POST data to '/RUapi/business' with the business's attributes.
 - addCategoryPage.js –
 - setAddButtonListener – listen for the 'add' button.
 - makeaddFunction(addButton) – AJAX POST to '/RUapi/category'
 - addItemPage.js
 - setAddButtonListener – listen for the 'add' button.
 - makeaddFunction(addButton) – AJAX POST to '/RUapi/items'
 - allBusinessesPage.js – Create the grid/list of businesses.
 - Uses AJAX to GET business data in JSON, assign parsed data into vars, then fill the table in display with fillTable
 - fillTable(name, address_line_1, address_line_2, city, zip_code, website, phone, id) – append list of data to the site to be displayed.
 - setTheOnClickListeners(id) – listen for click from array
 - addFunctionForSquare(thisSquare, id) – set a member of the clickListenersForSquares[] to a particular function.
 - allCategoriesPage.js – Create the grid/list of categories.
 - Uses AJAX to Get categories data in JSON, assign parsed data into data, then fill the table in display with fillTable.
 - fillTable(name, category_id, id) - append list of data to the site to be displayed.
 - setTheOnClickListeners(id) – listen for click from array
 - addFunctionForSquare(thisSquare, id) – set a member of the clickListenersForSquares[] to a particular function.
 - allItemsPage.js – Create the grid-/list of items.

- Uses AJAX to Get categories data in JSON, assign parsed data into data, then fill the table in display with fillTable.
 - fillTable(name, category_id, id) - append list of data to the site to be displayed.
 - setTheOnClickListeners(id) – listen for click from array
 - addFunctionForSquare(thisSquare, id) – set a member of the clickListenersForSquares[] to a particular function.
 - **loginPage.js – Send request to API with user credentials and return back a value.**
 - login() – Get values ‘username’ and ‘password’ from page. Send XMLHttpRequest() to the API. If there is a match, send user to the /AdminSite2/main.php with session. Send error if the user does not exist.
 - **singleBusinessPage.js – Similar as allBusinessPage.js above**
 - Uses AJAX to GET a specific business along with attributes using an ID. Uses ‘/RUapi/business/’ to get the data.
 - fillTable(name, address_line_1, address_line_2, city, zip_code, website, phone, id) – append list of data to the site to be displayed.
 - **singleCategoryPage.js – Similar as allCategoriesPage.js above.**
 - **singleItemPage.js – Similar to allItemPage.js above.**
- **addBusinessPage.php**
- **addCategoryPage.php**
- **addItemPage.php**
- **allBusinessesPage.php**
- **allCategoriesPage.php**
- **allItemsPage.php**
- **loginPage.php**
- **main.php**
- **singleBusinessPage.php**
- **singleCategoryPage.php**
- **singleItemPage.php**
- **Css**
 - **bootstrap.css – v3.3.5 – General page styling.**
 - **customStylesheet.css – Customized page styling for the website.**
 - **jumbotron-narrow.css – Customized headers, footers, containers.**
 - **map.css – has a width and height of 100% for #map.**
 - **media.css – Sets the navbar image and btn-lg.**
 - **publicSite.css – Sets styling for such things as navbar, headers, footers, containers, images, dropdown*, list*, contact*.**
- **HomeSite**
 - **about.php – Purpose and Donors**
 - **Css – bootstrap.css, jumbotron-narrow.css, publicSite.css, map.css**
 - **js – sponsorsFunct.js -> addLongSponsorsThanks()**
 - **img – bigstockphoto_Footprint.jpg**

- header.php, footer.php
- **business.php – Renders map with all the businesses. Lists all the businesses with contact and service. initBusinessMapWrapper(), sends a business name**
 - Css – bootstrap.css, jumbotron-narrow.css, publicSite.css, map.css, <https://cdnjs.cloudflare.com/ajax/libs/material-design-iconic-font/2.2.0/css/material-design-iconic-font.min.css>
 - js – mapFunc.js, listFunc.js -> addBusinessContact(name), addBusinessServices(name).
 - header.php, footer.php
- **category.php – Map businesses via category**
 - Css - bootstrap.css, jumbotron-narrow.css, publicSite.css, map.css
 - header.php, footer.php
 - js – mapFunc.js, listFunc.js
- **contact.php – List of contacts**
 - Css – bootstrap.css, jumbotron-narrow.css, publicSite.css, map.css, <https://cdnjs.cloudflare.com/ajax/libs/material-design-iconic-font/2.2.0/css/material-design-iconic-font.min.css>
 - img – CSCLogo.png, CSCRepairFair2.jpg
 - header.php, footer.php
- **error.php – Shows error in class ‘error-container’**
 - Css - bootstrap.css, jumbotron-narrow.css, publicSite.css
 - header.php, footer.php
- **footer.php – Footer Info**
- **header.php – Displays dropdown menus.**
 - js – jquery-1.11.1.min.js, headerFunc.js
- **home.php – Displays a key, about page, and sponsors information**
 - img – CSCLogo.png,
 - Css – bootstrap.css, jumbotron-narrow.css, publicSite.css, map.css
 - header.php, footer.php
 - js – mapFunc.js, sponsorsFunc.js
- **HomeSite.html – Displays map of area as well as content and navigation buttons.**
 - img – CSCLogo.png,
 - Css – bootstrap.css, jumbotron-narrow.css, publicSite.css, map.css
 - header.php, footer.php
 - js – mapFunc.js, sponsorsFunc.js
- **item.php – List items on a map**
 - Css – bootstrap.css, jumbotron-narrow.css, publicSite.css, map.css, <https://cdnjs.cloudflare.com/ajax/libs/material-design-iconic-font/2.2.0/css/material-design-iconic-font.min.css>
 - img – CSCLogo.png
 - header.php, footer.php
 - js – mapFunc.js, listFunc.js
- **recycle.php – Display businesses that recycle as a list and on a map.**

- Css – bootstrap.css, jumbotron-narrow.css, publicSite.css, map.css, <https://cdnjs.cloudflare.com/ajax/libs/material-design-iconic-font/2.2.0/css/material-design-iconic-font.min.css>
- img – CSCLogo.png
- header.php, footer.php
- js – mapFunct.js, listFunct.js

- **img**



- bigstockphoto_Footprint.jpg



- CSCLogo.png



- CSCLogoR.png



- CSCRectangular.png



- CSCRepairFair1.jpg



- CSCRepairFair2.jpg



- FB-f-Logo_blue_50.png



- Twitter_Logo_Blue.png

- js

- **Bootstrap.min.js - Bootstrap v3.3.5 - web page stylizing.**
- **BusFunc.js - Add/Update Business data.**
 - checkSession() – Make sure a session is active, if not, redirect to login page.
 - displayStates() - AJAX get for a list of 50 states, display 50 states in a drop down menu in the class 'form-control'.
 - addNewBusiness() - Get values from Business form, check for null in each entry and error checking. POST call to /RUapi/business with all the values.
 - displayTable() - Gets items from /RUapi/items into a list then adds the list of items with a checkbox.
 - clearAll() - empties the table, resets the addBusiness. Wipe screen to prevent clutter.
 - updateItem(value) - Add Business name and items using POST to /RUapi/updateBusiness
- **CatFunc.js - Add a new category to the DB.**
 - checkSession() - Make sure a session is active, if not, redirect to login page.
 - addNewCategory() - Check for form input is not null, else POST to /RUapi/category.
- **dbfunc.js -**
 - displayStates() - AJAX get for a list of 50 states, display 50 states in a drop down menu in the class 'form-control'.
 - checkSession() - Make sure a session is active, if not, redirect to login page.
- **headerFunc.js - Adds a drop down menu to the page for Repair, Reuse, and Recycle.**
 - addRepairLinks() - Creates a new XMLHttpRequest(), retrieves a JSON and places responseText into a list of items. Each in items sets an attribute link with item.php?cat=Repair%20Items&item="items[i]. Links are placed in document ID "header-repair-links". GET request is made to "/item/category/name/Repair%20Items"
 - addReuseLinks() - Creates a new XMLHttpRequest(), retrieves a JSON and places responseText into a list of items. Each in items sets an attribute link with category.php?name="items[i]. Links are placed in document ID "header-reuse-links". GET request is made to "/category/reuseExclusive".
 - addRecycleLinks() - Creates a new XMLHttpRequest(), retrieves a JSON and places responseText into a list of items. Each in items sets an attribute link with business.php?name="items[i]. Links are placed in document ID "header-recycle-links". GET request is made to "/category/recycleExclusive".
- **ItemFunc.js - Add items into the category.**
 - displayStates() - AJAX get for a list of 50 states, display 50 states in a drop down menu in the class 'form-control'.
 - checkSession() - Make sure a session is active, if not, redirect to login page.
 - addNewItem() - Get element by ID "name", check input is not null and give error if null, and POST the 'name' to the category to /RUapi/items'.
 - displayTable() - Gets items from /RUapi/category into a html table.

- `updateItem(category)` - Gets a 'name' from the console, POST 'name' and 'category' to '/RUapi/updateItems'.
- **jquery-1.11.1.min.js - jQuery v1.11.1. Functions and libraries to enable jQuery.**
- **listFunct.js – Script of functions formatting string and adding things to lists for display.**
 - `getFormattedPhone(phoneString)` – Formats a string of numbers to conventional styling of phone numbers.
 - `getFormattedZip(zipString)` – Formats a string of numbers according to conventional styling of zip codes.
 - `slashToUnderscores(string)` – Replaces the '/' with '_'.
 - `addItemList(catName)` – Takes name of category, gets the items from category to put into a list to be displayed at category-list-container.
 - `addBusinessList(categoryName, itemName)` – Takes name of category and a list of businesses to 'item-list-container'
 - `addBusinessContact(busName)` – Has the business name and contact info to the 'contact-container'.
 - `addBusinessServices(busName)` – Has a list of items a business accepts to the 'services-container'.
 - `addBusinessLinks(busName)` – Add a list of items to a business into the 'services-container'.
 - `printErrorMessage(titleClass, containerId, topic)` – Function for errors and debugging.
- **loginFunct.js – Verify correct login.**
 - `login()` – verification and encryption session startup.
 - `killSession()` – Kill session.
- **mainFunct.js – Page navigation.**
 - `checkSession()` - Make sure a session is active, if not, redirect to login page.
 - `addRoute()` – Directs user to correct add area.
 - `EditRoute()` – Directs user to correct search area.
- **mapFunct.js – Map functionality**
 - `pin(pinColor)` – google.maps pin creation.
 - `LatLng(lat, lng)` – Take latitude and longitude into a float and combine to dictionary.
 - `corvallisMap()` – set latitude and longitude for Corvallis Oregon, create a map object with google.maps.
 - `centeredMap (busLat, busLng)` – initializes the map centered on the given latitude and longitude.
 - `marker(latLng, map, pin, busName, busAddress, busCity, busState, busZip)` – Takes a businesses items as a dictionary object with google.maps.Marker.
 - `addInfoWindow(marker, map)` – Makes an informational window for a marker on the map.
 - `slashToUnderscore(string)` – replaces '/' with '_'.
 - `showErrorMap()` – Shows default map in case there is an error.
 - `initGeneralMap()` – Initializes a map with repair, recycling, and other businesses in three colors.

- `initCategoryMap(categoryName)` – Initializes a map with businesses from a given category, or all categories except Repair Items and Recycle if no category name is given.
- `initItemMap(categoryName, itemName)` – Initializes a map with businesses associated with a given category and item.
- `initBusinessMap(busName)` – Initializes a map with a pin for a single business with a given name.
- **registerFunct.js – Register a new user**
 - `checkSession()` - Make sure a session is active, if not, redirect to login page.
 - `newUser()` – Send an XMLHttpRequest() to create a new user.
- **searchBusFunct.js – Search for a Business**
 - `saveClicked()` – Get Elements by ID of business attributes, use AJAX to POST `‘/RUapi/changeBusiness’`
 - `searchBusiness()` – Use AJAX to GET data from `‘/RUapi/business/’`
 - `allBusinesses()` - Use AJAX to GET data from `‘/RUapi/business/’`
 - `editBusinessHelper(name)` – call `searchName` when one of the businesses is selected.
 - `delBusiness(match)` – Delete business by ID.
 - `getStatesDropdown(state_id)` – Use AJAX to GET states from `‘/RUapi/states/’`
 - `editBusiness()` – Use AJAX to GET business data from `‘/RUapi/business’` and name of business.
 - `editBusItems()` – Use AJAX to GET items of business from `‘/RUapi/items/’`.
 - `delBusItem(value)` – Use AJAX to DELETE item using business name and business item from `‘/RUapi/updateBusiness/’`
 - `updateItem(value)` - Use AJAX to POST item using business name and business item from `‘/RUapi/updateBusiness/’`
 - `allDocs(loc_id)` – Use AJAX to GET business docs from `‘/RUapi/businessdocs’`.
 - `addDoc()` – Adds a new document to the business using AJAX POST to `‘/RUapi/addBusinessDoc’`
 - `delDoc(doc_id)` – Deletes a document from a business using AJAX DELETE to `‘/RUapi/businessDoc’`
 - `clearAll()` – cleanup for new searches.
 - `checkSession()` - Make sure a session is active, if not, redirect to login page.
- **searchCatFunct.js – Search categories**
 - `searchCategoryByName()` – Use AJAX to GET category data from `‘/RUapi/category/’`
 - `searchCategoryByID(id)` – Use AJAX to Get category data by ID from `‘/RUapi/category/’`
 - `allCategories()` – Use AJAX to GET category data from `‘/RUapi/category/’`
 - `delCategory()` – Use AJAX to DELETE category by ID from `‘/RUapi/category/’`
 - `editCategory()` – Use AJAX to GET category data by name from `‘/RUapi/category/’`
 - `changeName(value)`

- `changeCategory()` – Uses AJAX to GET category data from `‘/RUapi/category/’`, make changes to data and then POST to `‘/RUapi/changeCategory/’`
 - **SearchItemFunct.js – Search Items**
 - `checkSession()` - Make sure a session is active, if not, redirect to login page.
 - `searchItem()` – Use AJAX to GET item data from `‘/RUapi/items/’`
 - `allItems()` – Use AJAX to GET item data from `‘/RUapi/items/’`
 - `searchItemHelper(name)` – Use AJAX to GET item data using the name from `‘/RUapi/category’`
 - `delItem()` – Use AJAX to DELETE item by ID from `‘/RUapi/item’`
 - `editItem()` – Use AJAX to GET item data from `‘/RUapi/items’` and prepare to make changes through `changeItem`.
 - `changeName(value)`
 - `changeCat(value)`
 - `changeItem()` – Use AJAX to POST item data to `‘/RUapi/changeItem’`
 - **sponsorsFunct.js – Donor page development**
 - `addShortSponsorsThanks()` – Renders a list of donors to display.
 - `addLongSponsorsThanks()` – Renders a list of donors to display.
- **Slim – slimframework.com, PHP micro framework to create APIs that consume, repurpose, or publish data..**
 - **Exception**
 - **Pass.php** – “This Exception will cause the `Router::dispatch` method to skip the current matching route and continue to the next matching route”
 - **Stop.php** – “Exception is thrown when the Slim application needs to abort processing and return control flow to the outer PHP script”
 - **Helper**
 - **Set.php**
 - **Http**
 - **Cookies.php**
 - **Headers.php**
 - **Request.php**
 - **Response.php**
 - **Util.php**
 - **Middleware**
 - **ContentTypes.php**
 - **Flash.php**
 - **MethodOverride.php**
 - **PrettyExceptions.php**
 - **SessionCookie.php**
 - **Environment.php**
 - **Log.php**
 - **LogWriter.php**
 - **Middleware.php**

- **Route.php**
 - **Router.php**
 - **Slim.php**
 - **View.php**
- **index.php -**