



POPSMARTKIDS.COM

NSC Application Development Team #1

NORTH SEATTLE COLLEGE

Team #1 of the North Seattle College
Applications Development AD 410.2346
Class, during the Spring 2018 quarter,
would like to present this proposal to:

PRIYANKA RAHA

Founder and CEO of PopSmartKids

Table of Contents

SECTION 1	3
Administrator Dashboard.....	3
1.1 User Manual	3
1.2 Administrator Login	3
1.3 Add New Administrator	3
1.4 Edit Administrator	3
1.5 Reset Administrator Password	3
1.6 Download Subscriber List	4
1.7 Delete User from Subscriber List.....	4
1.8 Administrator Logout	4
SECTION 2	4
Blog Dashboard.....	4
2.1 Create Blog Post.....	4
2.2 Edit Blog Post	5
2.3 Delete Blog Post	5
2.4 Blog Comments.....	5
SECTION 3	5
Product Dashboard.....	5
3.1 Add New Product	5
3.2 Add Product Image.....	6
3.3 Edit Product	6
3.4 Remove Product	6
SECTION 4	6
Google Analytics	6
4.1 Unique Site Visits	7
4.2 Pages Viewed Per Session	7
4.3 Bounce Rate	7
4.4 Top Landing Pages	7
4.5 Average Time Spent on Pages	8
SECTION 5	8
Code.....	8
5.1 The Makings of a Page	8
5.2 Home Page.....	9

5.3 Contact Page.....	9
5.4 Top.php.....	9
5.5 Bottom.php.....	10
5.6 Additional Pages	10
5.7 Uploading Images to Server	10
5.8 Change logo	10
5.9 Change favicon	11
5.10 Style and CSS Changes	11
SECTION 6.....	11
Entity Relationship Diagram	11
6.1 ERD.....	11
SECTION 7.....	12
MySQL Statements	12
7.1 Database Creation.....	12
7.2 Database Queries	15



SECTION 1

Administrator Dashboard

Note: You must be logged into the Admin Dashboard to perform any of the following actions pertaining to the Blog Page by visiting https://www.popsmartkids.com/admin/admin_login.php > Login

i 1.1 User Manual

Action: Access the electronic version of the user manual

Purpose: Access the e-manual via the website to utilize the search function and interactive features

How: Admin Dashboard > Admin User Manual (opens in a new tab for your convenience)

Note: You may also access, share or download the e-manual by navigating to the file path name directly by visiting [popsmartkids.com/manual.pdf](https://www.popsmartkids.com/manual.pdf)

i 1.2 Administrator Login

Action: Access the full admin dashboard

Purpose: Manage your blog page, products page, and database sections

How: visit [popsmartkids.com/admin/admin_login.php](https://www.popsmartkids.com/admin/admin_login.php) > Login

i 1.3 Add New Administrator

Action: Create a new administrator

Purpose: Add a new employee and grant privileges

How: Admin Dashboard > + Add Admin > Enter new user info and set password > Add Admin!

i 1.4 Edit Administrator

Action: Change information of admin user

Purpose: Add or remove an admin user's privileges, update their name or email address

How: Admin Dashboard > Edit Admin Data > click radio button next to said user > select Edit Admin button > update information > confirm by selecting Update Admin button

i 1.5 Reset Administrator Password

Action: Change an admin user's password
Purpose: Use this page when an admin needs a new password or has forgotten it
How: [Admin Dashboard](#) > [Reset Admin PW](#) > click radio button next to said user > select Edit Admin button > reset password > confirm by selecting Reset Password

1.6 Download Subscriber List

Action: Compile user email addresses into csv file
Purpose: Use this csv file to prepare all your user email addresses to send newsletters
How: [Admin Dashboard](#) > [Email Subscriptions](#) > [Download List](#)

1.7 Delete User from Subscriber List

Action: Remove email from your subscribers list
Purpose: To delete a user email address from your database
How: [Admin Dashboard](#) > [Email Subscriptions](#) > select DELETE to the right of said subscriber

1.8 Administrator Logout

How: [Admin Dashboard](#) > [Logout](#)



SECTION 2

Blog Dashboard

Note: You must be logged into the Admin Dashboard to perform any of the following actions pertaining to the Blog Page by visiting https://www.popsmartkids.com/admin/admin_login.php > Login

2.1 Create Blog Post

Action: Create a new blog post
Purpose: To post updates, news and other valuable information for users
How: [Admin Dashboard](#) > [Posts](#) > [Add New](#) > add info > Add Post

2.2 Edit Blog Post

Action: Edit a blog post

Purpose: Make changes to post Title, Author, Body content or image

How: [Admin Dashboard](#) > [Posts](#) > [Add New](#) > [add info](#) > [Add Post](#)

2.3 Delete Blog Post

It's never a good idea to totally "delete" something. Whether it be a blog post, a user who had opted out, or even unused photos. Deleting them means loss; they're gone forever. When you delete a post following the instruction below there is no confirmation or going back.

Action: Remove a blog post

Purpose: Make changes to post Title, Author, Body content or image

How: [Admin Dashboard](#) > [Five Most Recent Posts](#) > select "Post Disabled" next to said post

2.4 Blog Comments

Action: Take action on blog comments

Purpose: Either approve or deny users comments on your blog posts

How: [Admin Dashboard](#) > [All Pending Comments](#) (or by selecting "Comments" from right menu) > [Actions](#) > select either "Approve" or "Delete" for each pending comment



SECTION 3

Product Dashboard

Note: You must be logged into the Admin Dashboard to perform any of the following actions pertaining to the Blog Page by visiting https://www.popsmartkids.com/admin/admin_login.php > Login

3.1 Add New Product

Action: Create a new product

Purpose: Publish new product to Products page, visible by users

How: [Admin Dashboard](#) > [Products Dashboard](#) > [Add New](#) > [add info/images](#) > [Add Product](#)

Note: Remember to include the links to the Android/Apple store. Also, you cannot save the product until you have uploaded an image to the description. See Section 5.2 for instruction on adding images to a product before publication.

i 3.2 Add Product Image

Action: Add a new image to a product description

Purpose: Add a custom image to your products

How: [Admin Dashboard](#) > [Products Dashboard](#) > [Add New](#) > [Select image to upload](#) > [Choose File](#) > [select file from your computer](#) > [confirm upload by clicking Upload Image](#)

Note: You'll notice the Image name will update with your image title. If the upload is successful you'll be given an "Image Uploaded!" pop-up. If there is an error, you'll be prompted with the error message, "Sorry, only JPG, JPEG, PNG & GIF files are allowed." Try again. Also, you cannot upload the same image twice or you'll receive an error message.

i 3.3 Edit Product

Action: Change a product

Purpose: Make changes to the Product Name, Android/Apple links, Description, Image or keywords

How: [Admin Dashboard](#) > [Products Dashboard](#) > [Add New](#) > [Select image to upload](#) > [Choose File](#) > [select file from your computer](#) > [confirm upload by clicking Upload Image](#)

Note: Remember, you cannot save the product until you've uploaded an image to the description.

i 3.4 Remove Product

Action: Remove a product

Purpose: Remove a product from both the site and your database records

How: [Admin Dashboard](#) > [Products Dashboard](#) > [select "Delete" next to said Product](#)

Note: This removal is permanent. You will not be prompted to confirm the deletion.



SECTION 4 Google Analytics

Google Analytics is the most popular web analytic tool available today. Google Analytics lets you understand your site *and* users to better evaluate the performance of your content, products, and more. At no cost, you are able to process massive amounts of data quickly with an easy-to-use interface combined with shareable reports. Below are just a few of the reports created just for PopSmartKids.com. However, you can view [Google's full library here](#). Each of these searches can be performed from the Analytics Home page.

i 4.1 Unique Site Visits

Action: View how many people are visiting your site
Purpose: View user sessions and unique visitors to your site
How: Behavior > Overview > Select Unique Page Views from Dropdown Menu

i 4.2 Pages Viewed Per Session

Action: See what pages are viewed per session
Purpose: Perhaps the most valuable metric, this dives deeper into the user's experience than Unique Site Visits. Here, you can observe which pages are visited most, demonstrating which content engages more users.
How: Audience > Overview > Select Pages (or Sessions) Session from dropdown

i 4.3 Bounce Rate

Action: Learn the percentage of single page visits
Purpose: To see the number of visits in which a person leaves your website from the landing page, without browsing any further. Here, lower is better.
How: Behavior > Behavior > Overview > Bounce Rate is under Graph

i 4.4 Top Landing Pages

Action: View which page is "greeting" most customers
Purpose: Unlike the bounce rate, the Landing Pages Report will show you how people are arriving to your page. This report demonstrates which content is attracting the most users.
How: Behavior > Behavior Flow > displays Landing/Starting pages and their frequency

4.5 Average Time Spent on Pages

Action: The average time a user spends on a page

Purpose: It's important to understand that google measures this time differently. For example, if a hit is a "bounce" (meaning the user leaves immediately with no other site navigation) then no time is recorded. If not a bounce then know that google will continue counting the time, even on hidden tabs. This means your user could have "left" but because they haven't closed the tab, the clock continues.

How: Behavior > Behavior Flow > view Landing/Starting pages and their frequency



SECTION 5

Code

5.1 The Makings of a Page

How is a page created?

Your pages are comprised of multiple pieces. We use these pieces using `include` statements. Together these pieces come together as a "page template." This template is used throughout, and every single page follows this order. The template is comprised of a `top`, the `page` name, and the `bottom`, (and a backend `config` file dealing with database/MySQL statements) as seen below:



While the above is not an “actual” page on your site it’s a good example of how your pages are built. Notice how using smart code can be minimal and powerful. Let’s take a look at our `aboutUs.php` page:

```
<?php include 'includes/config.php'?>
<?php include 'includes/top.php'?>
<!-- START LEFT Column -->
    <div class="spacer"></div>
    <article>
        <h1>About PopSmartKids</h1>
        <div>
            <p>Content about who PopSmartKids.com is and what we do.</p>
        </div>
    </article>
<!-- END LEFT Column -->
<?php include ("includes/bottom.php") ;?>
```

#1: `top.php` entails the nav bar, your Beta button, logo, and others

#2: `pageName.php` includes everything seen here

#3: `bottom.php` includes the footer, your social media connections and other links

Note: You must have access to an IDE and an FTP client in order to perform any of the following actions pertaining to the files mentioned below. You must upload these files before changes are live.

Warning: Take great care when making *any* changes to these files. Only an experienced developer should tamper with these files.

5.2 Home Page

Action: Edit content on the homepage

Purpose: Change of the following sections:

- Flex slider images (see section 7.6 for instructions to upload images to your server)
- Content in the left and right columns

How: `root > index.php`

5.3 Contact Page

Action: Edit info on your Contact Us page

Purpose: Make changes to any of the following sections:

- The data you request from users such as name and email
- Feedback options you accept such as testimonial or comments
- ReCAPTCHA validator

How: While the “Contact Us” page is found at the root in `contactUs.php`, the real functions are found in `includes > simple.php`

5.4 Top.php

Action: Edit files in top.php file

Purpose: Make edits to any of the following sections:

- Navigation Bar
- Logo
- Beta Signup button

How: [includes > top.php](#)

Note: On the mobile version the “Beta” button is located at the root > index.php

5.5 Bottom.php

Action: Edit files in bottom.php file

Purpose: Make edits to any of the following sections:

- Footer
- Social Media icons

How: [includes > bottom.php](#)

5.6 Additional Pages

Action: Important information contained in additional pages

Purpose: To make changes to other sections not previously described

How [Employee section \(with bios and photos\) is located in root > aboutUs.php, and](#)
[Root > index.php contains the homepage YouTube links](#)

5.7 Uploading Images to Server

Action: Add images to the server

Purpose: To add images to your library so you can change things such as:

- Flex slider images
- Logo
- The “No Ads,” “No location tracking” (et al.) icons found on your Safe Zone page

How: [root > images folder](#)

5.8 Change logo

Action: Change the logo

Purpose: In the event you change the logo you’ll update the files on the server as well

- Be sure to upload the image to the server beforehand (see section 7.7)
- If you change the logo, be sure to change the favicon as well (see section 7.9)

How: Includes folder > top.php > header > nav > div class = "owl" > img src

5.9 Change favicon

Action: Change the logo

Purpose: The favicon is the small image viewable within the browser tab. It's a rendition of your logo and adds a further feel of customization to your site.

- Be sure to upload the image to the server beforehand (see section 7.7)
- If you change the favicon, be sure to change the logo as well (see section 7.8)

How: Includes folder > top.php > edit the favicon logo (You can also search for "favicon" within the file)

5.10 Style and CSS Changes

Action: Edit the css files

Purpose: Make changes to the formatting and styling of your pages such as:

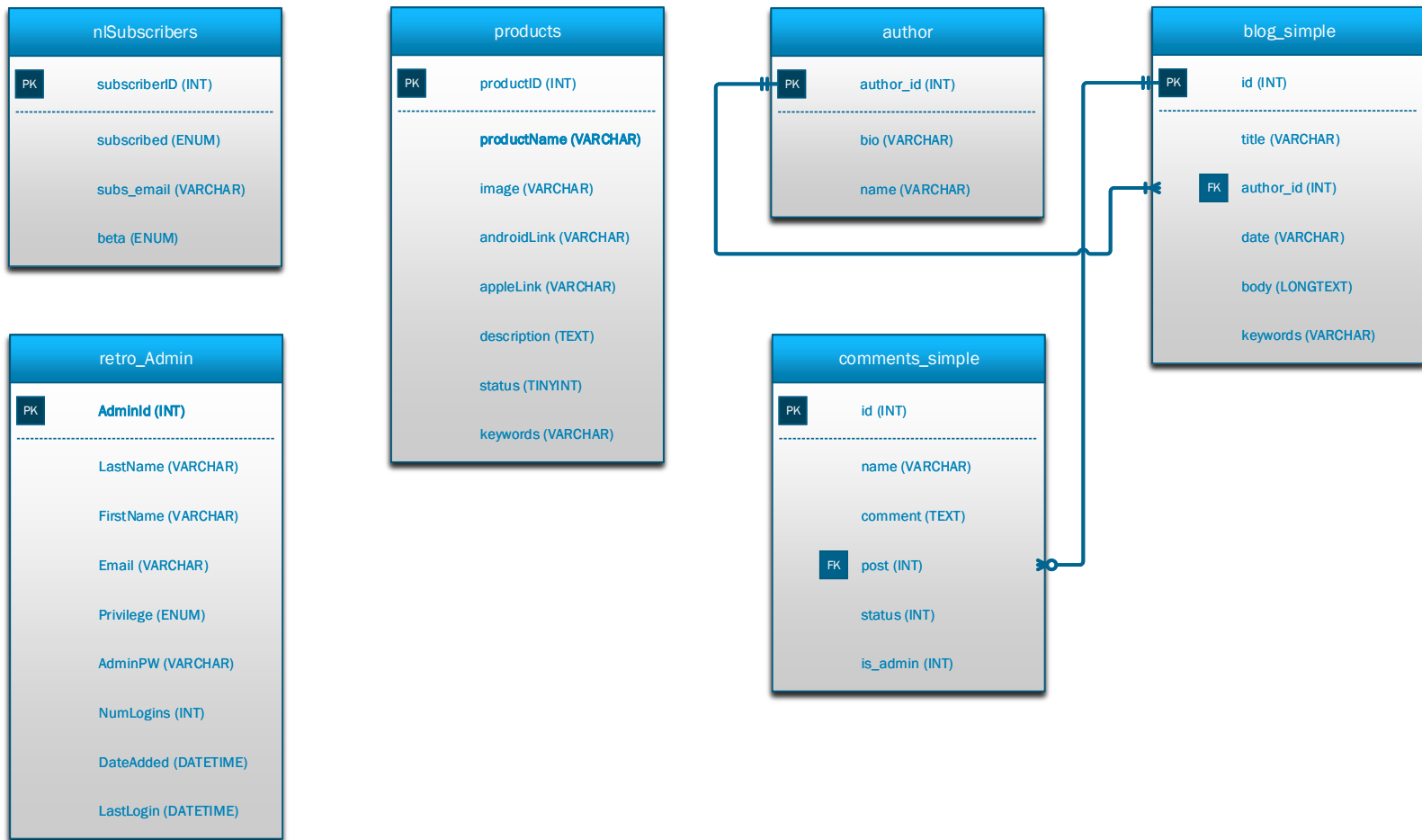
- form.css which controls your Contact page
- nav.css which controls your Navigation bar, which is featured at the top of each page, among others

How: root > css folder



|| **SECTION 6** Entity Relationship Diagram

6.1 ERD



SECTION 7

MySQL Statements

i 7.1 Database Creation

-- This code is constructed for the purpose and use of PopSmartKids.com as a backup in the event of a database loss. These MySQL statements should be handled with extreme care. Only an experienced professional developer should edit these or manipulate the database.

-- Published Saturday June 2, 2018 by Team #1 of the North Seattle College Spring 2018 quarter students of the AD410 Application Development Class.

-- -----
 -- Table structure for table `retro_Admin`

```
CREATE TABLE `retro_Admin` (
  `AdminID` int(10) UNSIGNED NOT NULL,
```

```

`LastName` varchar(50) DEFAULT '',
`FirstName` varchar(50) DEFAULT '',
`Email` varchar(120) DEFAULT '',
`Privilege` enum('admin','superadmin','developer') DEFAULT 'admin',
`AdminPW` varchar(255) DEFAULT NULL,
`NumLogins` int(11) DEFAULT '0',
`DateAdded` datetime DEFAULT NULL,
`LastLogin` datetime DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Indexes for table `retro_Admin`
ALTER TABLE `retro_Admin`
  ADD PRIMARY KEY (`AdminID`);

-- AUTO_INCREMENT for table `retro_Admin`
ALTER TABLE `retro_Admin`
  MODIFY `AdminID` int(10) UNSIGNED NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=9;
COMMIT;
-- end `retro_Admin`

-----
-- Table structure for table `nlSubscribers`
CREATE TABLE `nlSubscribers` (
  `subscriberID` int(11) NOT NULL,
  `subscribed` enum('Y','N') NOT NULL DEFAULT 'Y',
  `subs_email` varchar(60) NOT NULL,
  `beta` enum('Y','N') NOT NULL DEFAULT 'N'
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Indexes for table `nlSubscribers`
ALTER TABLE `nlSubscribers`
  ADD PRIMARY KEY (`subscriberID`);

-- AUTO_INCREMENT for table `nlSubscribers`
ALTER TABLE `nlSubscribers`
  MODIFY `subscriberID` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;
COMMIT;
-- end `nlSubscribers`

-----
-- Table structure for table `author`
CREATE TABLE `author` (
  `author_id` int(10) UNSIGNED NOT NULL,
  `bio` varchar(250) CHARACTER SET latin1 DEFAULT NULL,
  `name` varchar(100) COLLATE utf8_unicode_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

-- Indexes for table `author`
ALTER TABLE `author`
  ADD PRIMARY KEY (`author_id`);

-- AUTO_INCREMENT for table `author`
ALTER TABLE `author`
  MODIFY `author_id` int(10) UNSIGNED NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;
COMMIT;
-- End `author` table

```

```

-- -----
-- Table structure for table `blog_simple`

CREATE TABLE `blog_simple` (
  `id` int(11) UNSIGNED NOT NULL,
  `title` varchar(200) NOT NULL,
  `author_id` int(10) UNSIGNED NOT NULL,
  `date` varchar(100) NOT NULL,
  `body` longtext CHARACTER SET utf8 NOT NULL,
  `keywords` varchar(200) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Indexes for table `blog_simple`
ALTER TABLE `blog_simple`
  ADD PRIMARY KEY (`id`),
  ADD KEY `author_id` (`author_id`);

-- AUTO_INCREMENT for table `blog_simple`
ALTER TABLE `blog_simple`
  MODIFY `id` int(11) UNSIGNED NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=45;

-- Constraints for table `blog_simple`
ALTER TABLE `blog_simple`
  ADD CONSTRAINT `blog_simple_ibfk_1` FOREIGN KEY (`author_id`) REFERENCES `author`
  (`author_id`) ON DELETE CASCADE ON UPDATE CASCADE;
COMMIT;
-- end `blog_simple`

-- -----
-- Table structure for table `comments_simple`

CREATE TABLE `comments_simple` (
  `id` int(11) NOT NULL,
  `name` varchar(100) NOT NULL,
  `comment` text NOT NULL,
  `post` int(11) UNSIGNED NOT NULL,
  `status` int(1) NOT NULL DEFAULT '0',
  `is_admin` int(1) NOT NULL DEFAULT '0'
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Indexes for table `comments_simple`
ALTER TABLE `comments_simple`
  ADD PRIMARY KEY (`id`),
  ADD KEY `post` (`post`);

-- AUTO_INCREMENT for table `comments_simple`
ALTER TABLE `comments_simple`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=42;

-- Constraints for table `comments_simple`
ALTER TABLE `comments_simple`
  ADD CONSTRAINT `comments_simple_ibfk_1` FOREIGN KEY (`post`) REFERENCES `blog_simple` (`id`)
  ON DELETE CASCADE ON UPDATE CASCADE;
COMMIT;

```

```
-- End `comments_simple`

-----
-- Table structure for table `products`

CREATE TABLE `products` (
  `productID` int(10) NOT NULL,
  `productName` varchar(60) NOT NULL,
  `image` varchar(60) DEFAULT 'images/appicon1.png',
  `androidLink` varchar(60) DEFAULT NULL,
  `appleLink` varchar(60) DEFAULT NULL,
  `description` text,
  `status` tinyint(1) NOT NULL,
  `keywords` varchar(255) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

-- Indexes for table `products`
ALTER TABLE `products`
  ADD PRIMARY KEY (`productID`),
  ADD UNIQUE KEY `productName` (`productName`);

-- AUTO_INCREMENT for table `products`
ALTER TABLE `products`
  MODIFY `productID` int(10) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=72;
COMMIT;
-- end `products`
```

7.2 Database Queries

```
-- The following are useful queries that you will use often:

-- Selecting all blogs, and their authors
SELECT title, body, date, keywords, name, id
FROM blog_simple JOIN author ON blog_simple.author_id = author.author_id
ORDER BY id DESC

-- Updating a blog, where '$id' is the blog's ID number you wish to update
UPDATE blog_simple
SET title=:title, author_id=:author, body=:body, keywords=:keywords
WHERE `blog_simple`.`id` = $id

-- Permanently deleting a blog and any attached comments
DELETE FROM blog_simple WHERE id = $id

-- Approving a comment. "status" represents the display status of a comment,
--   where 0 is not to be displayed, and 1 is to be displayed.
UPDATE comments_simple SET status = '1' WHERE id = $id

-- rdy_DB() prepares a connection to the server. It grabs the databases host,
--   user name, password, and database name, then opens a PDO connection
--   to the server. PDO is an extra layer of security and functionality
--   that helps websites better interface with SQL servers. The open
--   connection will then be passed to whatever method needs it, and closed
--   automatically once the operation is complete.
function rdy_DB(){
```



```

//connection info for testing purposes
$servername = DB_HOST;
$server_username = DB_USER;
$server_password = DB_PASSWORD;
$dbname = DB_NAME;
$dsn = "mysql:host=$servername;dbname=$dbname;charset=utf8";
try{
    $db = new PDO($dsn, $server_username, $server_password);
    $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    //FOR DEBUGGING ONLY
    //echo "<p>Connected</p>";
} catch (PDOException $e) {
    $error_message = $e->getMessage();
    echo "<p>Error connecting to database: $error_message </p>";
    exit();
}
return $db;
//return null

```