**Readme File** The HTML website

The Website is for a bank called Professional Bank of Ireland (PBOI). It is designed using CSS stylesheets, HTML, Javascript and PHP

To view the site click on the file called **bankhome2.html** which is the **index/Homepage**

**USER VIEW**

The **user** will see five pages in the Website and they are all linked together internally through the navigation bar. The page links are

Home

Customers

Loans

Accounts

Personnel

These pages are all functioning connected webpages but the primary content of the site for now and for this assignment is contained on the Customers page. The other pages remain to be developed later but also as an illustration of the use of CSS as a feature of page design used across multiple pages.

Unfortunately I am still grappling with the concepts and execution of the update example features which should be in the website so I do not have a working model at the moment. I hope to be able to do this at some stage in the future.

**Home**  = Homepage

The **starting page** for the website is called **bankhome2.html** . Internally it is called Home **.** I deliberately did not use the word index for this page as it created complications while I was trying to learn and use other learning resources to experiment with. The picture on this page is .jpeg format. The navigation bar on this page is at the bottom of this page rather than the top (as on the other CSS styled pages) as example of inline/in page styling.

This is connected to **POIBpage2**.php and can be accessed through the navigation bar on the homepage on the Customers link.

**Customers = POIBpage2.php**

The primary content for this assignment is contained on this page. Embedded in the html page there is an **INSERT**form in php for client use to insert information into the appropriate table through the server. This has been tested and is working. Also on this page below the INSERT form(HTML & PHP), there is another embedded form table (The **SELECT**php page) for viewing returning information from the updated table. This incorporates features of HTML, PHP and Javascript form validation.

The form **validation** has been done for the first two fields of the form together using a variable function in Javascript form validation to prevent the form from being submitted if both or either of the first two form fields are left empty. An alert box is produced and instructions for which field(s) are empty. The empty boxes are also outlined in red. Another feature here is a third column added to the form field using HTML supplying an example of formatted data for entry in each field. Originally I put this in as a temporary measure in place of validation. I realised while doing the form that there would not be enough space for extended dropdown menus or radio button which were in the original plan.

**Reflection & differences from design document**

A problem with this form and its corresponding table in the db is the size of it. I have obviously made it far too big and thus created unnecessary difficulties for myself. However, I did not want to have incomplete records being inserted, as the dats fields would then be inconsistent. It seems a bit too late now to redesign the entire data base, though I can see that would be a good option. Lesson learnt=KIS (Keep it simple). I will try to cut down on the size of other tables in the db so that it will be more manageable for the finished product. I will have to remove columns to do this, and perhaps revamp some of the foreign keys which I can now see are not all likely to be user friendly for the client requirements. Obviously Rome wasn’t built in a day, and further practice would improve performance and output here. The assignment has been useful in clarifying for me the purpose and connections between the large range of different design elements and functions along with choices and considerations encountered in modelling, designing and producing even a very small functional database.

**Loans, Accounts and Personnel**

These pages currently (as indicated above) have no content on them but are retained as an illustration of CSS.

**CSS**

The page name is POIB.css. in the list of website files.

The features used for page design and styling were for

Background colour, navigation bar, font, headings and footer.

**Some of many Resources used**

**Online**

Javascript <https://www.youtube.com/watch?v=_Z-0cwONN6c>

HTML [www.W3Schools](http://www.W3Schools)

Form Validation with Javascript

<https://www.youtube.com/watch?v=Pc2e2YpKArg&spfreload=10>

<https://www.youtube.com/watch?v=xvXtb7mwMd8>

**POIB Scenario** **PROJECT TASK SHEET as given**

Professional Bank of Ireland (POIB) is a large financial banking group operating in both the republic of Ireland and Northern Ireland. It offers an extensive range of services to all its customers and its head office in in Dublin.

Members of the public can enquire about the bank services either by phone, correspondence or by calling into any of the bank branches. Having received the required information, those who wish to avail of these services are requested to supply certain personal and financial details to the bank. These potential customers will normally have to wait for a period of 48 hours, while the financial details are sent to the Credit Rating Section in the Headquarters in Dublin. If rejected, the personal details are recorded in a **rejection file**, and those members of the public receive a rejection note.

Clients wishing to apply for a bank loan must first fill out an application form and either mail it or deliver it by hand to any branch of the POIB. On receipt of the application for, the client’s details are confirmed using the **client file**. Incorrect client details are returned to the client for clarification. Approved members of the public have their personal details recorded in the client file. When client details have been confirmed, the client’s current balance is checked in the **Current A/C File**. If the client has other loans out at present, the loan A/C file is checked to see the total sum on loan at present. If either of these are unsatisfactory, the client is rejected. Otherwise, the loan details are recorded in the **loan A/C file**.

**Personnel** Section from time to time request details of staff who currently have loans out. Staff details are held in the Staff File, and the loan details are held in the Loans A/C File. The requested information is passed on to the Personnel Section. Personnel are responsible for the recruitment of all staff, including their interviewing and training. When new staff are taken on, Personnel Section pass on their details and the **Staff File** is updated. Likewise, any changes in staff details are passed on and updated in the Staff File.

New product brochures are produced every month and these brochures and sent to the home addresses of all the clients. From time to time, the Marketing Dept send us(?) a questionnaire for distribution to all the clients. Returned questionnaires are passed on to the Marketing Dept for analysis.

**Assumptions**

1. The database is required by the POIB HQ based in Dublin and is a central repository for details of customers nationwide. It will also store details of its customers in N.Ireland. The bank operates in both Eire and N. Ireland but only with accounts held in euros. This type of centralisation will inevitably produce a very large database which is likely to be quite unwieldy.
2. All POIB staff members become customers having their salaries paid into POIB. They will therefore be recorded in both the personnel table and the customers table which causes more duplication as they are already recorded in the personnel files. This will be augmented again when/if they apply for a loan.

Some of the entities in the db I designed has renamed some of the entities in the scenario above.

1. The rejection file has been replaced by a one row inclusion within the customers table.
2. The Staff File has been subsumed into a table called Personnel to reduce duplication, since Personnel holds personal records of staff members. This section is also responsible for HR issues including the hiring, training and deployment of new staff.
3. The Client File has been renamed ‘customers’ as in the customers table.

**Problem 1** : Currently there is a file system in operation which incurs a lot of isolation of data, duplication and inefficiency.

e.g 1. Customer details of all those applying to open any account with the POIB are recorded and stored, whether or not the potential customer is accepted.

**Solution 1**: A relational db will reduce the isolation and duplication of customer data through the use of primary and foreign keys on related tables.

The rejection file listed in the given scenario above will be eliminated by retaining all customers (both approved and rejected) on the ‘customer table’. The fields ‘customer status’ (approved, rejected or pending) and ‘sdate’(status date) will supply the details. A customer who is rejected now, may in the future seek or be invited to reapply to become a customer. It is unlikely that the number of rejected customers would be high but staff would have direct access to this information.

Customers can be resident in N.Ireland or Eire but the main office is in Dublin. For new accounts, personal details are supplied to the branches and registered in the Client file. New customers have to wait for 2 days to see if i they are approved as new customers. Their details have already been recorded during this waiting period but their status is pending until they are either approved or rejected as new customers. If rejected, they are notified as such, but their records are still kept by the bank where the rejection is recorded again in a rejection file.

**Solution**- Record new customer details directly on to Client database tables and update one field of the record to update change of status.

Customers must have current account, details of which must be checked and confirmed to be eligible for a new loan account.

**Solution**-Keep one set of records only and include the branch no. and name of county in the client file to identify location as either Eire or N.Ireland

Personnel and Staff File which should be operating in conjunction with each other appear to be currently isolated as the Personnel Section send out requests for updates on staff who currently have loans out.

**Solution**- Allow Personnel access to view records for themselves of those staff members with existing or pending bank loans. As all staff members are already recorded in Personnel records, it would seem more efficient.

**Solution** - They need to function as one unit. Personnel should maintain and update **all** not **some** staff records. Current A/C and Loan A/C are working in isolation even though they depend on each other for up to date data records. Access to both these sets of data needs to be aligned into one set of records, accessible to both sections.

**County**

The inclusion of the column ‘county’ in both staff and customer tables is included to show whether staff and customers respectively are based in in Eire or N Ireland.