**TASK09 - SEND A MESSAGE**

**Design Details**

Note: started on this revised design (which I thought would be quick) to add outgoing messages to its capabilities. Since it was NOT quick and this is not a first priority, it is more important that I stay ahead of the people now working. Therefore, finish the other designs, plan unit testing, and manage our developer first.

1. References:
   1. Current version of our ‘Website Standards yymmdd.docx’
   2. Current version of Create and Drop yymmdd.txt – our DDL. Rename to .sql.
   3. Current version of Insert and Delete yymmdd.txt – our DML. Rename to .sql
2. Objective:
   1. Create a simple input form to send a message, ask questions, or send an email.
3. Description:
   1. A single button will be added to the website in the “Our Work” section on the “Join In” page at the end of the first paragraph titled, “Researchers”.
      1. The new button will be placed to the left of the existing button.
      2. It will be identical to the existing “About grants” button except:
         1. Background color, in accordance with Ref 1.a. above is palevioletred.
         2. Inscription: “Contact us”
   2. Clicking the button opens a small message box on top of the existing page for the user to send us a message.
      1. The box will be draggable around the screen by the user. Please let us know if this is an issue.
      2. Is there an issue with pop-up blockers blocking the opening of the box?
      3. Possibly use a Meta Box for this.
4. The database DDL & DML code (Ref 1.b. & 1.c.) for all website database tables including the message tables is provided separately.
5. Data elements:
   1. The Message table will have columns as in the attached DDL statement. Of those columns:
      1. Msgdate is a timestamp automatically populated by the database.
      2. Insubjectcode will be user selected from a dropdown menu.
      3. interestlevel should be omitted and left null for future use.
   2. Flask Form input data elements:
      1. Name – entered by the user
      2. Email – the user’s email address entered by the user. This email address will not be validated other than to the extent it is in a valid email format.
      3. Message – completed by the user
6. On the pop-up form:
   1. The top title will say “Send in a message or question”
   2. On the form the user will click for a pull-down menu of subjects. Of these one and only one must be selected.
      1. For the pull-down list, one item will be displayed for each row returned from table ‘messagesubj’ where ‘menusequence’ is not null and ‘active’ is True.
      2. The messages will be displayed in ascending order by ‘menusequence’ (from the messagesubj table).
      3. The title above the subject list will say “Select a subject”
      4. The system will record the ‘insubjectcode’.
7. The Send a Message input form:
   1. The current version of our Website Standards, Ref 1.a., should be followed in all respects.
   2. Except for Ref 1.a. (which takes precedence), the form will appear similar to the Contact Us page in the About section in terms of fonts, colors, field descriptions, etc. except the font size may need to be smaller.
   3. Please include a description for each field like in the MyIdea input form (accessible by clicking “Share My Idea” in the “Grants” section of the website). Note: this form is being replaced in a future task so generally please do not use it as an example.
   4. *All fields on the “Send a Message” form will be validated to assure acceptance by the database before submission.*
   5. A “Submit” button is placed at the bottom of the “Send a Message” form. Clicking this will update the database and display the message “Thank you for your message.” A small “x” in the upper right corner or other method should be used to allow/require the user to acknowledge and close the “Send a Message” box.
8. Include in the “About” section on the “Contact Us” page a duplicate of the pop-up box but displayed permanently rather than as a pop-up. Please also use the background color already on the page.
9. Send emails for specific messages:
   1. A trigger will be added to table ‘inmessage’.
      1. Note: the website already includes an automated process for sending emails. It is in the Results section on the Questionnaire page at the end of the last paragraph (see questionnaire\_cont.html in \templates\content\ for the relevant html). The logic and process for sending this email can be used in creating this trigger.
      2. After a row is inserted into table ‘inmessage’ the trigger will:
         1. Use ‘insubjectcode’ from the new row, access table ‘insubjectcodeemails’ and return all rows for that ‘insubjectcode’. These provide the email addresses (‘destuseremail’) that should receive the forwarded message each time an incoming message is received for that ‘insubjectcode’.
         2. Must track through each data element in the following steps:
            1. Assure all data elements needed are listed on table ‘outmsgque’.
            2. Track back to the ultimate source, each of these data elements.
            3. Reflect the flow of the data element in the above instructions & language.
         3. Use ‘insubjectcode’ from the new row, access table ‘inmessagesubj’ and return ‘shortdesc’ (saved as ‘messagesubj’) and ‘subjectdesc’ for the ‘insubjectcode’.
         4. Insert one row into table ‘outmsgque’ for each row on table ‘subjectcodeemails’ for this ‘insubjectcode’.
      3. The second part of the original trigger coded for the ‘inmessage’ table will be modified and installed on table ‘outmsgque’ to send an email for each row inserted into ‘outmsgque’. The trigger will:
         1. Get the new row from table ‘outmsgque’.
         2. The email will be as follows:
            1. Subject: Automated SFF message - <shortdesc>.
            2. Body: “SFF received the following message:

Date: [formatted as mm/dd/yyyy hh:mm AM]

Email subject: <subjectdesc> [from table ‘inmessagesubj’]

From: <username>

Sender email: <useremailaddress>

Message: <message>”

* + - * 1. In smaller print at the bottom: ‘This message was from an unmonitored mailbox.’

1. Please deliver to and unit test in GitHub. You will not need to install them on our production server as we will incorporate them in a future official version release of the website.
2. All previous versions of this document are invalidated and replaced hereby.