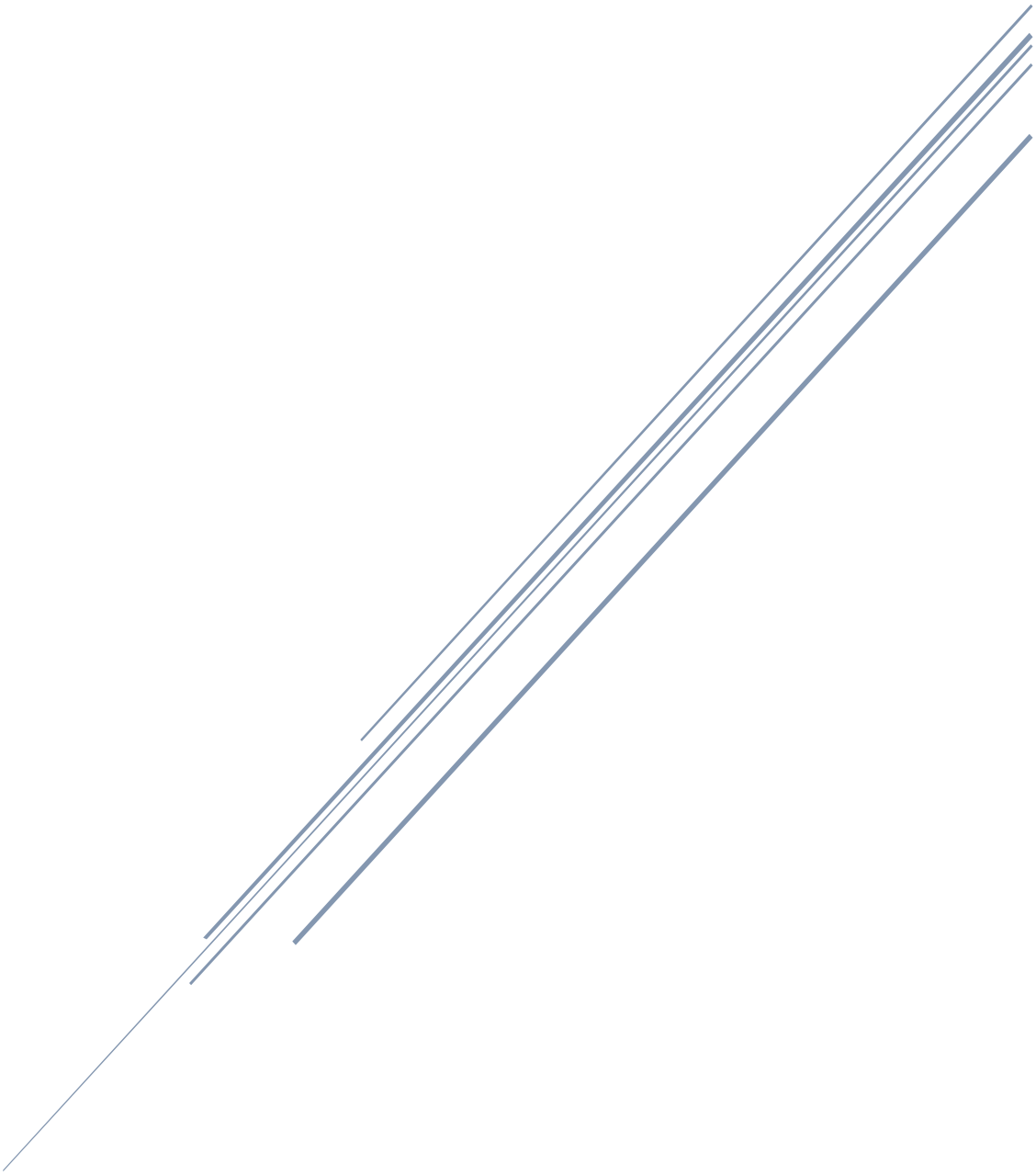
TECHNICAL DOCUMENTATION – ASK ME



Group 1

Online Voting and Polling application

Contents

Ask Me – An Overview

The Ask Me Application was designed to allow a conference session leader to invite their audience to vote on questions that are posed to them by the leader. It can be used across a multitude of devices from mobile phones to laptops and desktop PCs. It supports a projector view and is built upon a variety of different programming and scripting languages. To work on this application, it is advised the developer has a strong background with AJAX, PhP and JSON.

Programming, Scripting and Mark Up languages Used

- AJAX
- Bootstrap
- Bootstrap Admin Template
- JavaScript
- HTML5
- CSS
- Chart.JS
- MySQL
- PhP
- JSON

Front End Webpages

conferenceModify.php

The conferenceModify.php page is built to enable the user to change their conference name after it has already been created. It consists of AJAX calls to the database to get the elements from the database and replace the conference name with the new name.

conferenceSelection.php

The conferenceSelection.php page is built as a landing page after the user logs in. It will allow the user to create a conference, create a new user (if the logged in user has administrator rights) and to view all created conferences. It consists of an AJAX call to update the conferences on page load, as well as an update button to update conferences on the fly.

forgotpassword.php

The forgotpassword.php page is a simple form that will allow a user to submit their email address via an AJAX call to the database, to retrieve their security question. Once the question is returned, the user can answer their question to then enter a new password.

login.php

The login.php page contains a AJAX call to the database to check and validate the user as they enter their details, as well a link to create a new account. Once logged in, the user will be asked to accept or decline the terms of use.

logout.php

The logout.php page clears the JSON file stored for the session as the user logs out of the application.

new_conference.php

The new-conference.php page will allow the user to create a new conference, it uses an AJAX call to the database to submit the fields the user has entered. This will submit the row to the conferences within the database to add the new conference.

new_question.php

The new_conference.php page will function in a similar way to that of the new conference page, it uses an AJAX call to the database to submit the fields that are entered by the user. This will then add a row to the database table.

new_session.php

The new_session.php page will function the same as the new_conference.php and the new_question.php pages. It consists of an AJAX call to submit data to a database.

presenter.php

The presenter.php page consists of an AJAX call that continuously calls to the database to look for an open question, once a question is open it will then present the screen with the current open question.

registration.php

The registration.php page consists of a form for user registration, it is to be used by an administrator to create accounts for the audience to sign into when using the system for the first time. The page uses an AJAX call to commit a row to the database to enter a new user.

results_visual.php

The results_visual.php page is used to view the results in a graph form using Chart.js. The page contains AJAX calls to the database to collect and manipulate the results from the database and turn them into an array that can be input into the dataset of the chart.

session.php

The session.php page is the landing page of the session. It consists of an AJAX call that continuously calls to the database to check for an open question, once one is opened and the database is updated, the screen will update to the currently open question for the users to vote on.

session_admin.php

The session_admin.php page is the administrator page for a session, it allows the admin to create questions, open a question or view results of a closed question. This is all done with AJAX calls to the database to collect the data, or to commit the new data.

sessionModify.php

The sessionModify.php page works in the same way that the conferenceModify.php page does. It will allow the user to submit a new session name via AJAX and update it in the database.

sessionSelection.php

The sessionSelection.php page is designed to send an AJAX call to the database to retrieve a list of sessions to display on the screen. It also contains the option to create a new session, as well as the option to modify the conference.

Back End Includes

account_creation

create_account.php

The create_account.php include contains PhP code to create and submit a user to the database. It also contains back-end validation to check for incorrect passwords, poor email format and incorrect characters.

conference_creation

create_conference.php

The create_conference.php contains PhP code to create a new conference and commit it to the database. It contains back-end validation to check for any errors, such as incorrect start or end date, blank fields or a bad name.

conference_manage

get_conferences.php

The get_conferences.php page contains PhP code to populate the page with the conferences already created within the database.

pull_data.php

The pull_data.php contains PhP code to get the conference data from the database and display it on the page

store_id_in_session.php

The store_id_in_session.php contains PhP code to store the current conference within the JSON session for use within other pages.

update_conf_name.php

The update_conf_name.php contains PhP code to retrieve the current conference name depending on the conference stored in session, and then PhP code to update the conference name within the database.

database_conn

db_connection.php

The db_connection.php contains PhP code to connect to the database, it contains the log on information for the database.

login

forgot_password_verify.php

The forgot_password_verify.php contains PhP code to check that a user has entered the correct answer to their recovery question and then update a field in the database to allow the user to enter a new password.

forgot_verify.php

The forgot_verify.php contains PhP code to allow the user to enter a secret answer to a security question. This will allow the user to recover their account if they forget their password in the future. The code will submit the answer to the database.

login_validation.php

The login_validation.php page contains PhP code to check the username and passwords are correct. This will allow the user to log in to the application. The PhP code contains back-end validation of the input fields.

update_password.php

The update_password.php file contains PhP code to allow a user to add a new password to their account after they have answered their recovery security question.

validate_user_cred.php

The validate_user_cred.php file contains PhP code to check that the users log in details are correct, if they are not the session will be destroyed and restarted. The user will be directed to the login page.

question_creation

create_question.php

The create_question.php file contains PhP code to create a new question, it will submit the question into the database as a new row upon completion.

get_sessions_conference.php

The get_sessions_conference.php file contains PhP code to retrieve the list of sessions within a conference from the database to be displayed on the screen.

question_manage

get_questions_new.php

The get_questions_new.php file contains code to retrieve the questions relevant to the session from the database.

open_close_questions.php

The open_close_questions.php file contains PhP code to check if a question is currently marked as open in the database, and if it is not, it will mark it as open. If the question is currently marked as open, it will alert the user that a question is already open.

results

get_results.php

The get_results.php file contains PhP code to retrieve the results of a question based on the question ID stored in the session. It will group the responses to summarise how many people vote on which option.

get_type.php

The get_type.php file contains PhP code to set the type of graph chosen within the application.

results.php

The results.php file contains PhP code to post the question ID to the session.

store_type.php

The store_type.php file contains PhP code to store the selected graph type from the results page.

session

check_sessions_open.php

The check_sessions_open.php contains PhP code that will look for open questions within a session.

get_conf_name.php

The get_conf_name.php contains PhP code that will retrieve the conference name from the database.

get_session_name.php

The get_session_name.php contains PhP code that will retrieve the session name from the database and store it in the JSON.

get_time.php

The get_time.php file contains PhP code that will get the time from the server.

submit_answer.php

The submit_answer.php file contains PhP code that will send the users submitted answer to the database for storage.

session_creation

create_session.php

The create_session.php file contains PhP code that will allow the administrator to create a session within a conference and store it within the database. It will also submit the session to the JSON. The file contains back-end validation to check for incorrect characters, invalid time and invalid dates.

get_conferences.php

The get_conferences.php file contains PhP code that will retrieve the conferences from within the database.

session_manage

get_sessions.php

The get_sessions.php will get the session information from the current conference.

store_session_in_session.php

The store_session_in_session.php file contains PhP code to get the session information from the database and store it within the local JSON file.