A Problem Domain

I have got a good experience during a coding process. Along the implementation I realised what I do not have a clear understanding of the problem domain and I need to work with unfamiliar libraries and constructs. Even additional research on each individual problem didn’t bring an ideal result for my project as almost every problem had several different solutions and it wasn’t easy to choose the right one because of the lack of experience. At the end of the development stage I could clearly understand the problem domain and if I would need to make another similar system or to work on another version of the current, I would build it in a better and more efficient way. I would like to repeat my words what it was a great experience as I had chance to learn on my own mistakes.

List of the problems and solutions:

1. **Materialize (HTML/CSS/JS) framework is used in the basement of the website what is providing some components and plugins. It gave me a chance to build a website faster, but was causing problems when I had to integrate my own elements or when I had to change the elements provided by the framework. I have noticed what Materialize framework is a quite young project and there are almost no any tutorials and information about it online. Some related problems I had to deal with are documented below:**

* ***Sliding gallery wasn’t taking the rest height of the screen after the navigation bar. It was working fine on large screens and had a broken output on small screens.***

**Materialize framework has three screen sizes (Mobile Devices <= 600px, Tablet Devices <= 992px, Desktop Devices > 992px) to help with building responsive designs. The height of the navigation bar is different for each screen size and to programmatically fit the sliding gallery into the entire screen height I had to add some code what was getting the height of the navigation bar after the page was loaded and pass that value to the sliding gallery plugin for a further calculation of the height required for the sliding gallery.**

* ***The solution of the problem above caused another problem. Plugin loading speed wasn’t fast enough and it was giving a confusing output on slow internet connections.***

**Problem was solved by adding a preloader what is displayed until the plugin is completely loaded on a page.**

* ***HTML forms input elements, loaded with AJAX, weren’t recognising their values what were loaded as well. That input elements were working as expected after they were focused.***

**Problem was solved by programmatically focusing/ blurring all the form elements and scrolling a page to the top. That was giving a feeling what page is loaded as expected.**

* ***Materialbox plugin of the Materialize framework wasn’t working as expected and was causing some internal library errors.***

**Problem was solved after some testing of plugin elements. A data-cation attribute of img tag was a reason of the problem when it contained an empty string as a value. That value was dynamically added from a database and I had to replace empty value with a whitespace.**

1. **Problems I had to deal with working with AJAX:**

* ***New elements loaded with AJAX weren’t responding on any actions.***

**To solve the problem helped a Chrome DevTools where I found what new elements didn’t have any event listeners attached to them. After the reason was discovered it was easier to search for a solution. To every new element of a page loaded with AJAX, new event listeners need to be attached after they are loaded and the frameworks plugins need to be called to have an effect on that new elements.**

* ***It was necessary to pass some extra data to a server together with the form data on the form submission.***

**After some research I discovered a formData object (what is providing a key/value pair construct) and a jQuery ajax() function. The complete action of solving a problem was:**

* + **Prevent default browser form submission after submit button was clicked.**
  + **Create a new formData object with form key/value pairs.**
  + **Append new key/value pairs to the formData object.**
  + **Send the formData object to a server using ajax() function.**
* ***AJAX request was failing with an error message “php” is not found when a request was sending to a “php tasks/phpscript.php”.***

**I gave a “php tasks” name to a folder using a whitespace between the words. AJAX saw just a first word of the path and was trying to find a php script with the same name as this word. I renamed the folder replacing the whitespace with an underscore symbol and that solved the problem.**

1. **A MySQL problem I had to deal with is documented below:**

* ***Iteration through the query result object to get a data didn’t give any result like it was empty. Testing showed what query was successful and contained some records from a database.***

**After deep analysis of the code I have noticed what I was iterating through the same query object twice. The first iteration was giving an expected result and the second one was ignored and didn’t provide any output. After a couple of hours of web surfing with a hope to find a solution I have discovered some interesting information. Query object has a special pointer what is increased when you iterate through the object and if it is necessary to iterate through that object again the pointer need to be reset to 0 using the mysql\_data\_seek($result, 0) function.**