Deployment guide

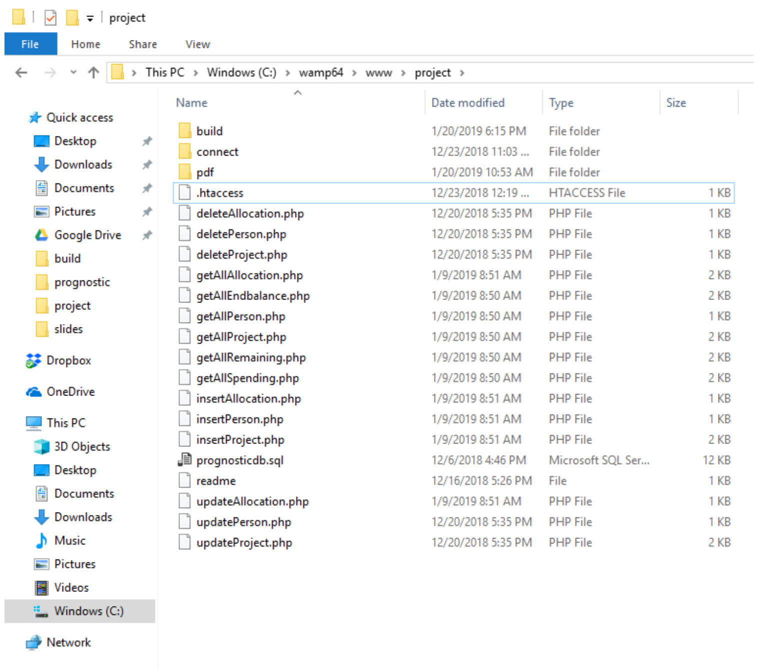
This guide will explain how to deploy the React application into an application that can be uploaded and executed by a server. We will also mention known issues and design flaws.

1. Open prognostic/src/components/allfunctions.jsx with an editor
2. Change the BASE constant to:
   ```
   const BASE = "../"
   ```
3. Navigate to the prognostic folder through the CMD/Terminal, and execute:
   ```
   npm run build
   ```
4. In the prognostic/build folder, open the file index.html with an editor.
5. The paths to JavaScript-files and CSS-files need to be prepended with a dot, for example:

   ```
   <link href="./static/css/1.b4a35927.chunk.css" rel="stylesheet">
   <script src="./static/js/main.ad1c4339.chunk.js">
   ```

   The build process don't prepend the links with a dot for the `href` and `src` attributes.
   NOTE: This must be done for all CSS-files and JavaScript-files that index.html references.

6. Change the name of index.html to index.php
7. Copy the build folder and place it as it is in the server where the PHP-files are located. The folder structure should look like the following:



This PC › Windows (C:) › wamp64 › www › project ›

Name	Date modified	Type	Size
build	1/20/2019 6:15 PM	File folder	
connect	12/23/2018 11:03 ...	File folder	
pdf	1/20/2019 10:53 AM	File folder	
.htaccess	12/23/2018 12:19 ...	HTACCESS File	1 KB
deleteAllocation.php	12/20/2018 5:35 PM	PHP File	1 KB
deletePerson.php	12/20/2018 5:35 PM	PHP File	1 KB
deleteProject.php	12/20/2018 5:35 PM	PHP File	1 KB
getAllAllocation.php	1/9/2019 8:51 AM	PHP File	2 KB
getAllEndbalance.php	1/9/2019 8:50 AM	PHP File	2 KB
getAllPerson.php	1/9/2019 8:50 AM	PHP File	1 KB
getAllProject.php	1/9/2019 8:50 AM	PHP File	2 KB
getAllRemaining.php	1/9/2019 8:50 AM	PHP File	2 KB
getAllSpending.php	1/9/2019 8:50 AM	PHP File	2 KB
insertAllocation.php	1/9/2019 8:51 AM	PHP File	1 KB
insertPerson.php	1/9/2019 8:51 AM	PHP File	1 KB
insertProject.php	1/9/2019 8:51 AM	PHP File	2 KB
prognosticdb.sql	12/6/2018 4:46 PM	Microsoft SQL Ser...	12 KB
readme	12/16/2018 5:26 PM	File	1 KB
updateAllocation.php	1/9/2019 8:51 AM	PHP File	1 KB
updatePerson.php	12/20/2018 5:35 PM	PHP File	1 KB
updateProject.php	12/20/2018 5:35 PM	PHP File	2 KB

8. In your browser, type:
   localhost/project/build/index.php

Known issues

There exist some problems and design flaws with the application that is good to know about, I will elaborate on the ones we ha noticed in this section.

Navigation

There is some problem between the navigation after deploying and uploading the application to an apache server. The navigation works, but the different tabs in the application is not files that apache know about, hence if one tries to refresh the page apache can't find it.

Save file

The save file functionality is implemented and working, but the navigation problem also causes the save file not to load properly. When the button is pressed, the URL is updated in the browser, but the PHP files is not loaded, but if one select the address field and manually press ENTER it loads.

Global save / frontend-backend interface

The interface between the frontend and backend is problematic. The interface forces the front-end tabs to create unique ID's when adding new objects, and then the interface will change these ID's when the global save is pressed, making the ID's inconsistent (which means that the objects can't be targeted for an update or delete). Because of lack of time to attend to the problem, we added a forced reload of the page after the global save is pressed so the front-end gets the new ID's directly from the server.

Because the global save reloads the page, the navigation problems applies here also and making apaches display "page can not be found" (except if you press the global save when you are on the index page because that's the only page apaches knows about)

Infinite loop redraw

There is a known bug with the timeline component that can manifest if the browser is set on a certain zoom-percentage (its differ across computers). The timeline will render incorrectly and be laggy. There were many quick fixes to try but none did work (seems to depend on browser and version), to fix the problem more time than we had need to be invested.

SQL-injection

No safety/security is implemented on the backend and all queries that the server receives is directly sent to the database, hence SQL-injection problems.

Ajax-calls

All ajax-calls that is made is synchronous ajax-call which will perform poorly in a real environment. When the page loads, synchronous ajax-calls fetch the data and saves it locally and when its saved to the server synchronous ajax-calls sends it.

Destroyed components on tab-change

All components is destroyed and recreated when navigating through the tabs, this cuases poor UX on the allocation tab because one will not be in the same place when navigating back to the component.