------------------ TECHNICAL OVERVIEW ------------------

This project(database) is completely developed in open source web technologies mentioned as follow

Front End   : HTML, CSS, JavaScript, jQuery
Back End   : PHP
Database   : MySQL
Author   : Kandarp Patel (kppatel14392@gmail.com)
Mentor   : Simon Stephan (simon.stephan@mv.uni-kl.de)
Department : Lehrstuhl für Thermodynamik, TU Kaiserslautern

I have tried to create overall project(database) with comments in each page for better readability.
However, Overall project overview is as follow.

------------------ PROJECT STRUCTURE ------------------

This project contains following files and folders. Purpose of all folders and files are described here.

Project Folder (Path) : moleculeDB/
Entry File   : index.php
Config File   : **config.php** (described in senario)
Root Files   : project master files which can be considered as main files, uses the other resources from following folders.

Folders (moleculeDB/)
- css/   ---> include global stylesheets of project
- **doc/**   ---> contains all documents related to this project
- fonts/   ---> custom downloaded font used by stylesheets for this project
- **function/**   ---> function which contains algorithms used throughout this project
- gen/   ---> include auto generated pm/ls files and logs.
- img/   ---> contains all images used in this project
- **include/**   ---> contains sub files, which are includes by root files
- js/   ---> all JavaScript files used in project
- mailer/   ---> e-mail library for email function
- test/   ---> test files which were created at the start of the project. not used in current working project

------------------ LIST OF DIPLAY FILES (FRONT END to BACK END) ------------------

To change static contents from any page in database, best way to track files from front end.

Steps:

- open database in browser
- find any content needs to be change
- read URL of browser (http://localhost/molecularDB/**mollist.php**?###)
- check which page is opened in browser
- go to that page (**ex. mollist.php**) from backend, find content that want to change
- if not found, check for the 'include' tag. (**ex. include 'include/footer.php'**) which contain some included file (ex. footer.php)
- open that included file (ex. footer.php) and find content
- change content

Note : for home page(http://localhost/molecularDB/)--> **'index.php'**
Detailed scenario with screenshots are described in separate file (**admin-scenario.php**)

Following are list of files, which are displayed in the browser. So it may help admin to edit static contents of website.

DISPLAYED PAGE	BACKEND PAGE
/* user pages */	
Home page	index.php/welcome.php
Molecule List	mollist.php
Molecule Detail	moldetail.php
Reference List	reflist.php
Nomenclature	normen.php
contact	contact.php
/* admin pages */	
New Molecule	addmol.php
New Reference	addRef.php
Update Molecule	updatemol.php
Delete Molecule	deletemol.php
Update Reference	updateRef.php
Delete Reference	deleteRef.php

------------------ FILE NAMING CONVENTION ------------------

overall naming convention of project files are as follow.

FILENAME PATTERN	PURPOSE
'##list'	file contain molecule list/reference list display
'update##'	file contain update algorithm for molecule/reference
'delete##'	file contain delete algorithm for molecule/reference
'add##'	file contain insert algorithm for molecule/reference
'process##'	all process files after user submit/user action
'##Func'	file contains function which used in project

throughout the project I have used some shortcuts in file name to understand what that file means.

ABBREVIATION	IDEA
'mol'	molecule
'ref'	reference
'norm'	nomenclature
'mst'	master data
'det'	detail data
'down'	download
'gen'	generate
'up'	update
'delete'	delete
'add'	insert
'head'	header

I think, mentioned information are enough to get idea about the purpose of file, by seeing filename

FOR EXAMPLE

1.processUpRef = process + Up + Ref
- process : contain logic after user submit some data
- Up : update logic
- Ref : reference data
- Overall idea : this file contain logic when admin request for update some specific reference

2.updatemol = update + mol
- update : update form
- mol : of molecule
- Overall idea : this file contain form of updating molecule

3.processUpdateHead: = process + update + head
- process : after submit form logic
- update : update logic
- head   : header data (reference or molecule - but reference don't have data in parts, molecule have -> header + detail)
- Overall idea : this file contain logic to process data after user submit from for update molecule header.

Overview of database structure and tables are described in separate file (2.database-overview.pdf)
Detailed scenario with screenshots are described in separate file (3.admin-scenario.pdf)

------------------ **THANK YOU**