Contents

[File Structure 2](#_Toc405815001)

[Customer Tables 3](#_Toc405815002)

[Account Details 4](#_Toc405815003)

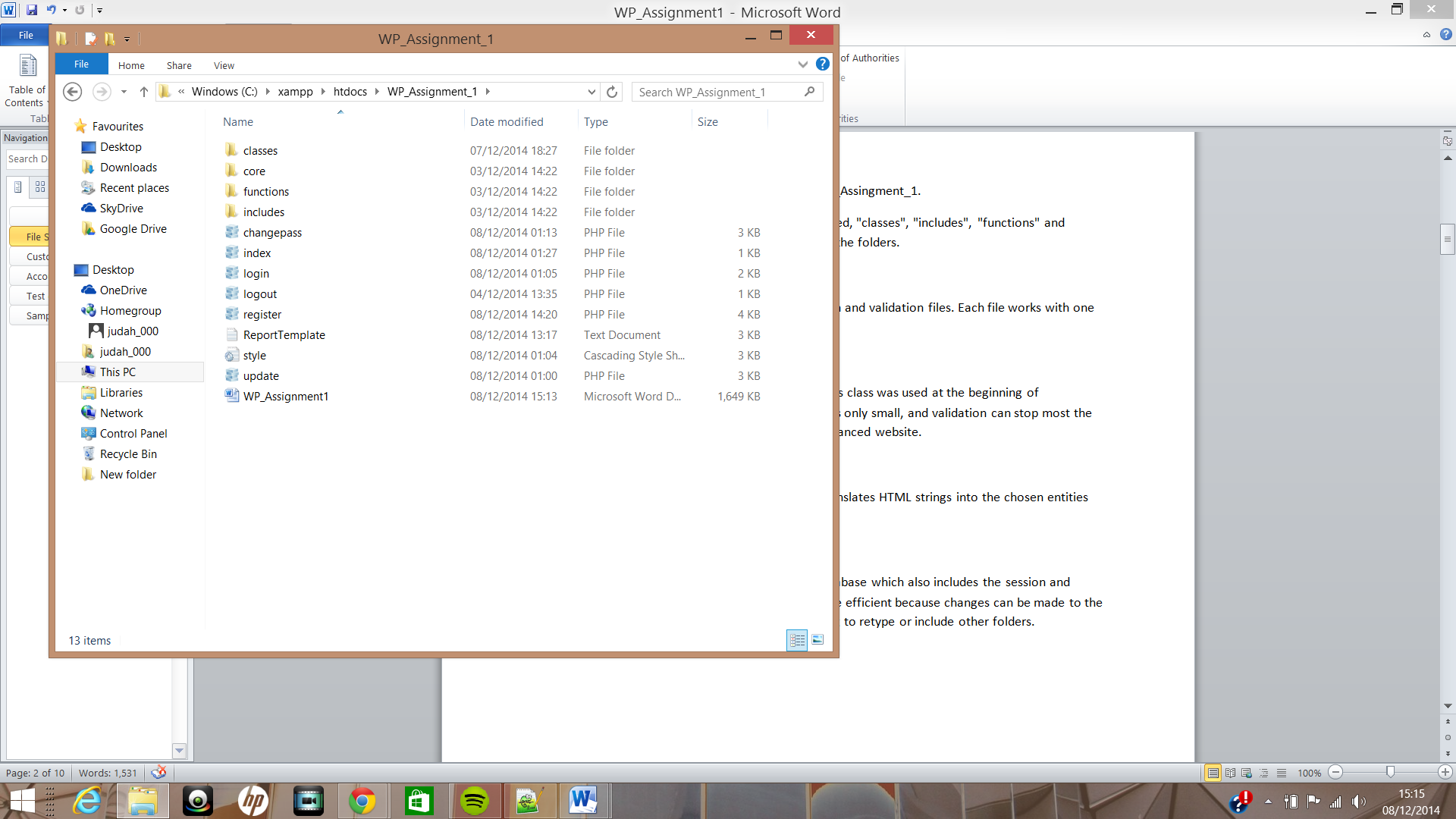
[Test cases 5](#_Toc405815004)

[Sample data 9](#_Toc405815005)

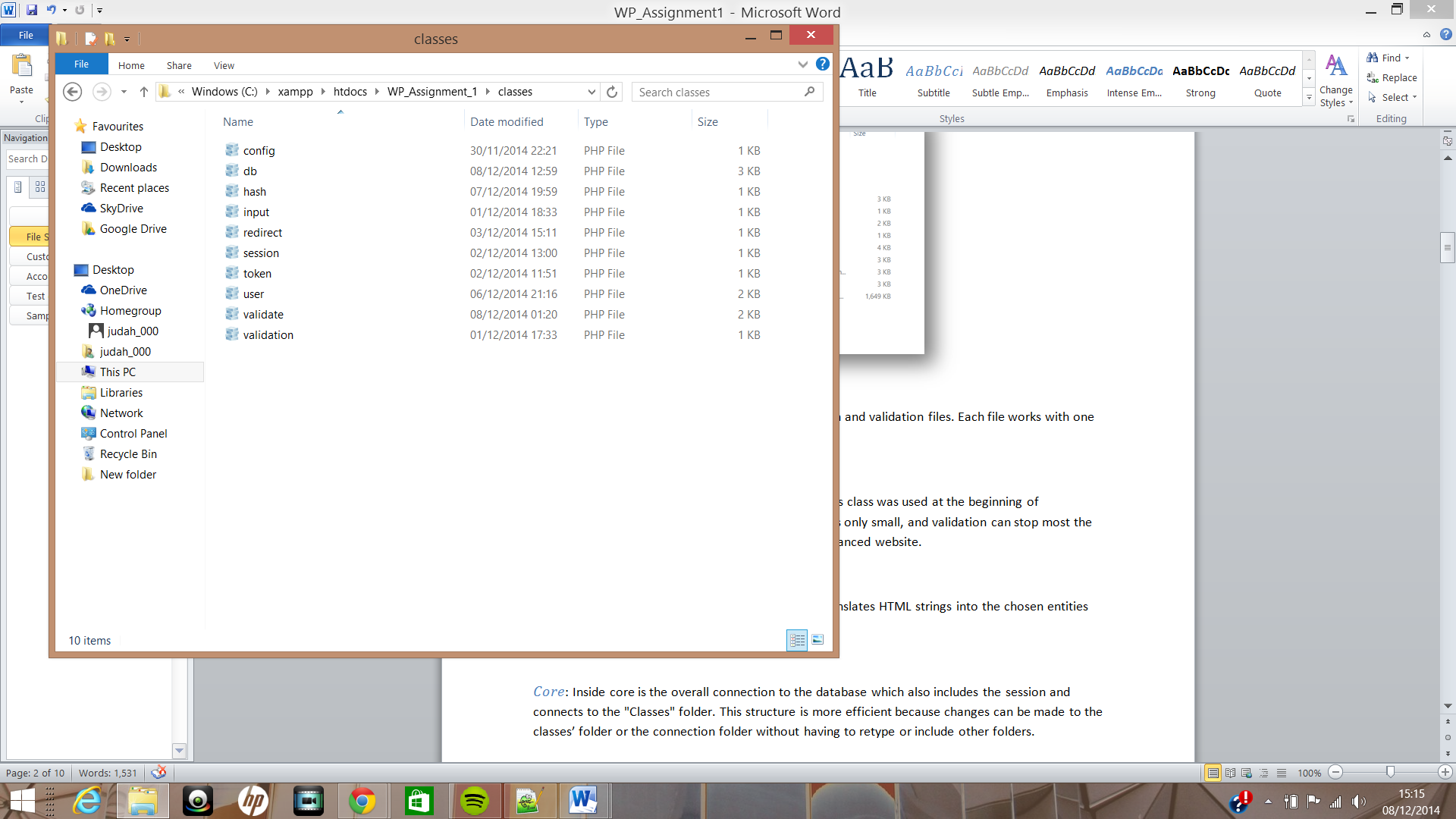
# File Structure

The whole system is located inside a folder called WP\_Assingment\_1.

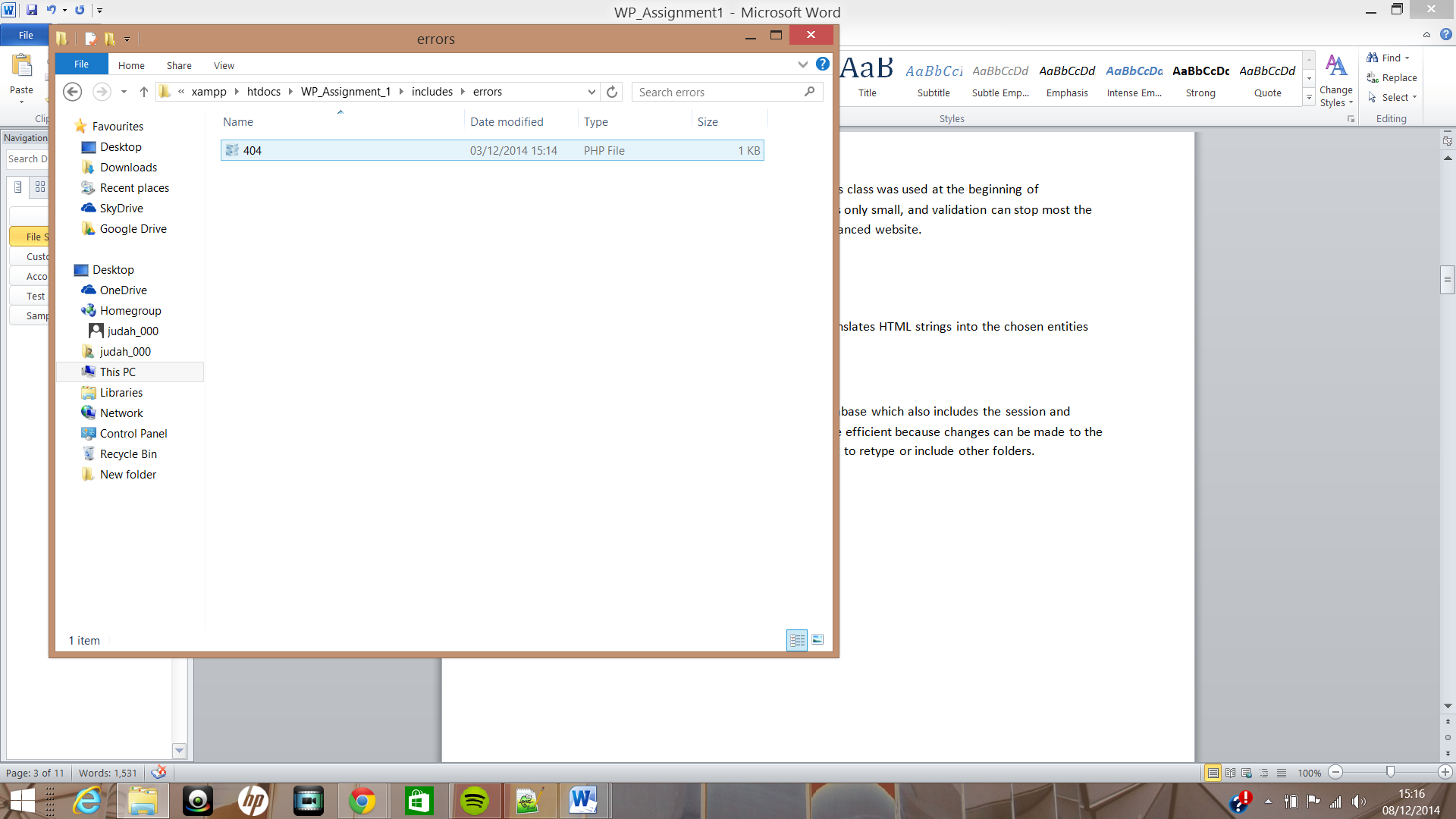
I separated the website into four different folders called, "classes", "includes", "functions" and "core". And the interactive pages were saved outside the folders.



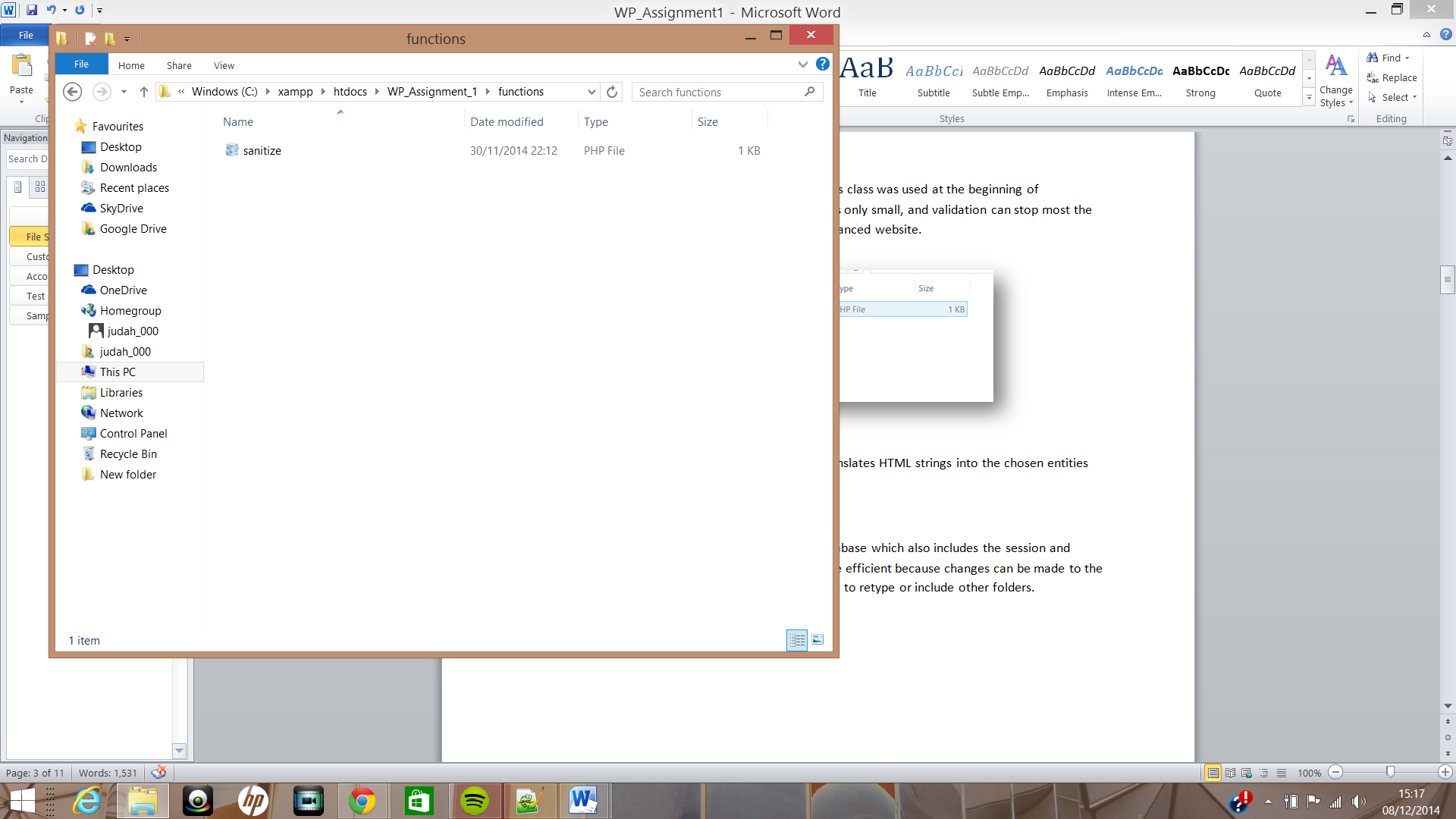
Classes: inside classes are security, database, session and validation files. Each file works with one another to ensure the user is safe while logged in.



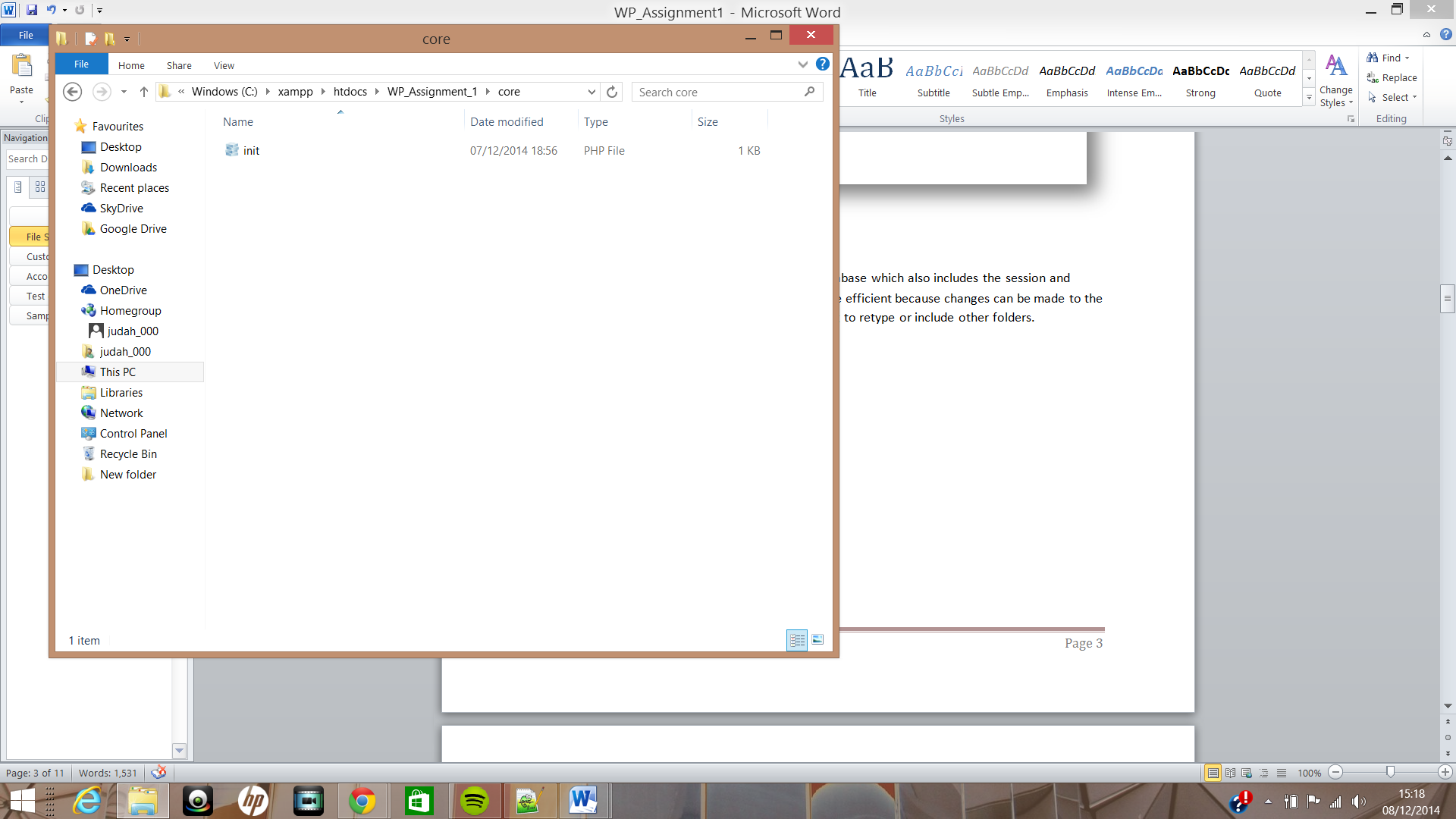
Includes: Inside includes is an error re-directory. This class was used at the beginning of development but I chose not to use it as the website is only small, and validation can stop most the errors. This file can be used when creating a more advanced website.



Functions: Inside functions is a sanitize file. This translates HTML strings into the chosen entities (UTF-8). This makes the website more secure.



Core: Inside core is the overall connection to the database which also includes the session and connects to the "Classes" folder. This structure is more efficient because changes can be made to the classes’ folder or the connection folder without having to retype or include other folders.



# Customer Tables

CREATE TABLE IF NOT EXISTS `users` (

`id` int(11) NOT NULL,

`username` varchar(20) NOT NULL,

`password` varchar(64) NOT NULL,

`salt` varchar(32) NOT NULL,

`name` varchar(55) NOT NULL,

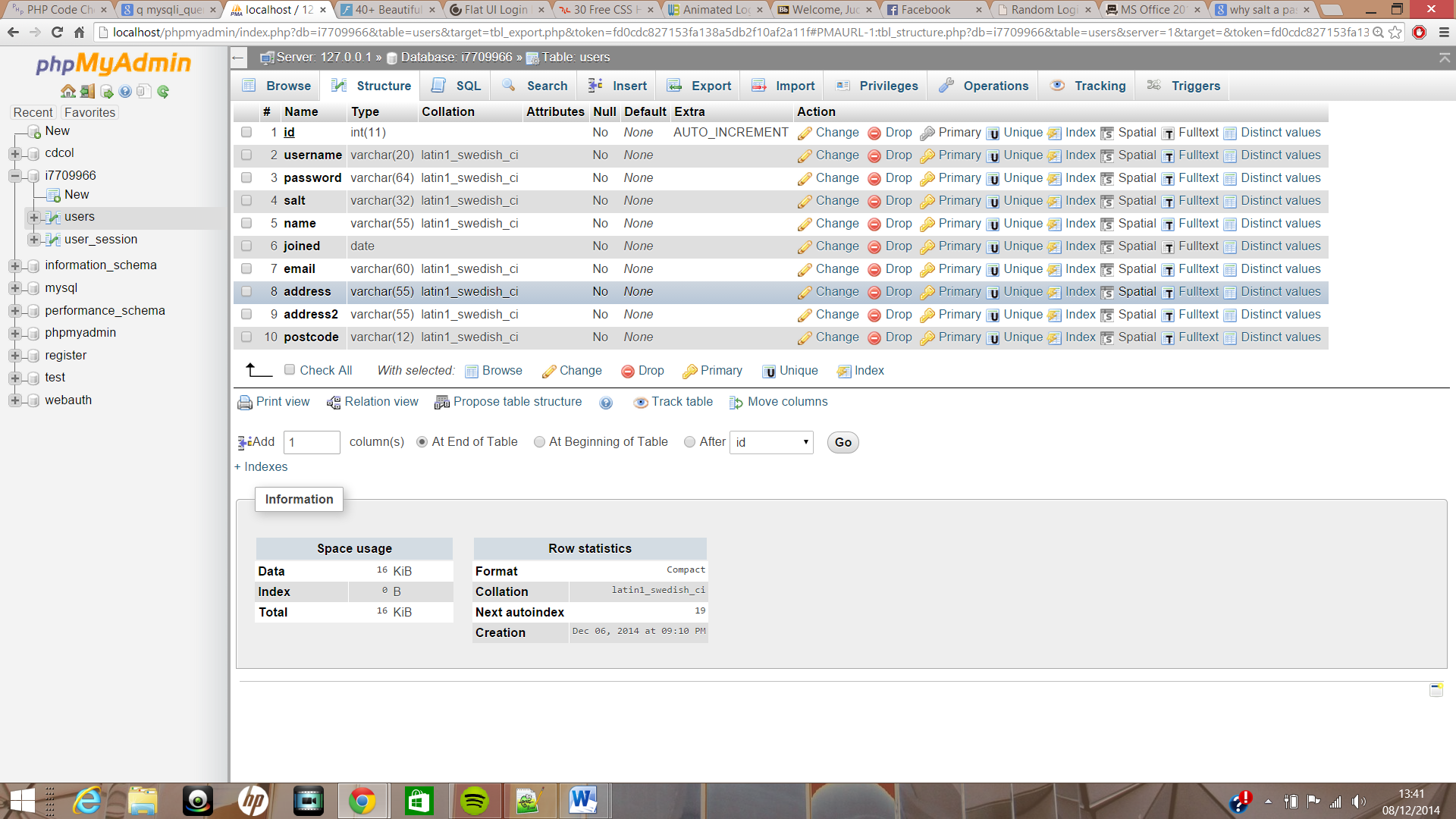
`joined` date NOT NULL,

`email` varchar(60) NOT NULL,

`address` varchar(55) NOT NULL,

`address2` varchar(55) NOT NULL,

`postcode` varchar(12) NOT NULL

 PRIMARY KEY (`id’))

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO\_INCREMENT=19 ;

I used one table with standard e-commerce details. Only one table was needed because data could be inserted by updating the table.

Salt is inserted into the database to hash the password, making it more difficult for malicious attackers to hack users.

# Account Details

WinSCP:

Host name: student20393.201415.uk

Username: i7709966

Password: Ott12345

MySQL:

Username: i7709966

Password: 069ca095950a9e5bd82ccfe422d74a06

Database: i7709966

# Test cases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Summary | Procedure | Expected result | Actual result | Status  (P)pass  (F)Fail |
| 1 | Check Register validation.  Repeat password does not match password | 1. Enter all details on register.php 2. Enter a password that doesn’t match password | Error message.  New user is not created | Error message. New user wasn’t created | P |
| 2 | Password min:6 | 1. Select register button 2. Enter details 3. Add password with 5 characters | Error message.  New user is not created | Error message. New user wasn’t created | P |
| 3 | Try to register user with no password | 1. Select register button 2. Enter details 3. Leave password fields blank | Error message.  New user is not created | Error message. New user wasn’t created | P |
| 4 | Register  Username min:2 | 1. Select register button 2. Enter details 3. Username : a | Error message.  New user is not created | Error message. New user wasn’t created | P |
| 5 | Register  Username max:20 | 1. Select register button 2. Enter details 3. Username with 60 characters | Error message.  New user is not created | Error message. New user wasn’t created |  |
| 6 | Register with no username | 1. Select register button 2. Enter details 3. Leave username field blank | Error message.  New user is not created | Error message. New user wasn’t created |  |
| 7 | Name min:2 | 1. Select register button 2. Enter details 3. Name:a | Error message.  New user is not created | Error message. New user wasn’t created |  |
| 8 | Name max:55 | 1. Select register button 2. Enter details 3. Name with 56 characters | Error message.  New user is not created | Error message. New user wasn’t created |  |
| 9 | Register with no name | 1. Select register button 2. Enter details 3. Leave name field blank | Error message.  New user is not created | Error message. New user wasn’t created |  |
| 10 | Email min: 5 | 1. Select register button 2. Enter details 3. Email with < 5 characters | Error message.  New user is not created | Error message. New user wasn’t created |  |
| 11 | Email max: 60 | 1. Select register button 2. Enter details 3. Email with > 60 characters | Error message.  New user is not created  Error message.  New user is not created | Error message. New user wasn’t created Error message. New user wasn’t created |  |
| 12 | Register without entering email field | 1. Select register button 2. Enter details 3. Leave email field blank | Error message.  New user is not created | Error message. New user wasn’t created |  |
| 13 | Register address field max: 60 | 1. Select register button 2. Enter details 3. Address field >60 character. | Error message.  New user is not created | Error message. New user wasn’t created |  |
| 14 | Register without address | 1. Select register button 2. Enter details 3. Leave address fields blank | Error message.  New user is not created | Error message. New user wasn’t created |  |
| 15 | Postcode max: 12 | 1. Select register button 2. Enter details 3. Enter > 12 characters on postcode | Error message.  New user is not created | Error message. New user wasn’t created |  |
| 16 | Postcode required for register | 1. Select register button 2. Enter details 3. Leave postcode fields blank | Error message.  New user is not created | Error message. New user wasn’t created |  |
| 17 | Register a user | 1. Select register button 2. Enter details 3. Press register | User created.  Details are in database.  Redirect to login.php | New user created and redirected to lgin.php | P |
| 18 | Login as an unregistered user | 1. Go to login screen 2. Enter invalid details 3. Press login | Error message.  Did not log in | Error message.  Did not log in | P |
| 19 | Login as registered user, incorrect password | 1. Go to login screen 2. Enter valid username 3. Enter wrong password 4. Press login | Error message. Did not log in | Error message. Did not log in | P |
| 20 | Login as valid user | 1. Go to login screen 2. Enter valid username 3. Enter valid password 4. Press login | Flash, welcome {username}, session created | User is logged into index.php. welcome message received. | P |
| 21 | Update details | 1. Login as valid user 2. Settings->my details 3. Enter new details | Flash, details updated.  Redirect to index.php.  New details stored in database | Redirected to index.php. flash message popped up | P |
| 22 | Update details validation. Name < 2 | 1. Login as valid user 2. Settings -> my details 3. Enter details 4. Name : a | Error message.  Details shouldn’t update | Error message.  New details did not update | P |
| 23 | Update details validation.  Name < 50 | 1. Login as valid user 2. Settings -> my details 3. Enter details 4. Name : 51 characters | Error message.  Details shouldn’t update | Error message.  New details did not update | P |
| 24 | Update details validation.  Email < 5 | 1. Login as valid user 2. Settings -> my details 3. Enter details 4. Email : me@a | Error message.  Details shouldn’t update | Error message.  New details did not update | P |
| 25 | Update details validation.  Email > 60 | 1. Login as valid user 2. Settings -> my details 3. Enter details 4. Email: 61 characters | Error message.  Details shouldn’t update | Error message.  New details did not update | P |
| 26 | Update details validation.  Postcode > 12 characters | 1. Login as valid user 2. Settings -> my details 3. Enter details 4. Postcode: 13 characters | Error message.  Details shouldn’t update | Error message.  New details did not update | P |
| 27 | Change password | 1. Login as valid user 2. Settings -> change password 3. Enter details 4. Press change button 5. Logout 6. enter username 7. enter new password | Message. Password is updated in db | Message. New password | P |
| 28 | Change password validation  Password min:6 | 1. Login as valid user 2. Settings ->change password 3. Enter details 4. Password 5 characters | Error message.  Details shouldn’t update | Error message.  New details did not update | P |
| 29 | Change password validation new password matches new password again | 1. Login as valid user 2. Settings ->change password 3. Enter new password 4. Enter new password again incorrectly | Error message.  Details shouldn’t update | Error message.  New details did not update | P |
| 30 | Change password validation. Current password doesn’t match actual password | 1. Login as valid user 2. Settings ->change password 3. Enter invalid password | Error message.  Details shouldn’t update | Error message.  New details did not update | P |
| 31 | Check salt on passwords | 1. Go to register page 2. Enter details correctly 3. Press register button 4. Check database table on phpmyadmin | The password and salt fields should look encrypted. Salt hashed | Password field generated random character and salt is hashed | P |
| 32 | Check if token matches session | 1. Go to login page 2. Enter details correctly 3. Right click -> inspect element -> resources->cookies 4. Check the unique SessID | User is logged in |  | P |
| 33 | Check if user is still logged in with session | 1. Echo {id} when user logs in 2. Check if user ID matches ID in database talbe | Match | Match | P |
| 34 | When user logs out, session destroys | 1. Go to login page 2. Enter valid username 3. Enter valid password 4. Press login 5. Press logout | Redirect to login.php. |  | P |
| 35 | Session flash |  |  |  |  |
| 36 | Make sure each salt and hash is generated randomly so if two or more users have the same password they have different salt and hash | 1. Register a new user with password:123456 2. Register another user with password:123456 3. Check phpmyadmin 4. Look up users table 5. Compare the users password and salt | The same passwords should look different | They look different | P |

# Sample data

