# Report

## Breakdown List:

|  |  |
| --- | --- |
| parsing using PHPExcel Library | 3 hours |
| Extracting data and returning JSON data | 3 hours |
| Error handling and exception | 3 hours |
| HTML/ CSS and design, uploading data | 2 hours |
| Using Pi chart library | 3 hours |
| Parsing JSON data to Pi chart and display data | 1 hour |
| Creating database and user authentication | 2 hours |
| Using jCryption library, Encryption | 3 hours |
| User sessions and logout | 1 hour |

Total **15** hours

## Description:

Technologies used: PHPExcel, jCryption, DrawPieChart, CSS, HTML, JavaScript, jQuery, PHP and bootstrap

Most time consuming tasks are parsing, error handling and encryption.

* Parsing
  + System can detect wrong type of file
  + System can detect if file is corrupt or uploading errors
  + System can show error on empty files
  + System can detect if file extension is .xls or .xlsx but not valid file
  + System can show results if title sheet has no chart title on A1 index. It will name it ‘untitled’
  + System will work if A1 and B1 are other than ‘Count’ and ‘Name’
  + System will skip all non-numeric Count values
  + System can tell if data has no value
  + System will tell if there is no ‘data’ sheet
  + JSON data response returns after parsing
* Pie Chart
  + Pie chart shows labels on hover
  + Pie chart display values in percentage
  + Pie chart is svg and animated
  + Pie chart has always unique colors for every piece
  + pie chart change color on hover
* Encryption
  + Using jCryption openSSL library
  + OpenSSL RSA and AES for keys
  + Public and private key encryption
  + 4096-bit key for encryption
  + Creates good confusion and diffusion
* User Authentication
  + User authenticates against database (MySQL) record
  + Default user name is: demo
  + Default password is: admin
  + Creates login sessions that remains until user logs out
  + Access control to all pages

# Library used and reasons

## PHPExcel

* Well written and supported library for using Excel sheets in php
* Good exception handling
* Easy to use and documentation is available

## jCryption

We can use SSL. Using SSL certificate from Certificate authority ensures data integrity and data encryption. In more sophisticated way, we can avoid man in the middle attacks but its costly and not suitable for test projects.

* Use openSSL
* RSA for key exchange and AES for encryption
* Public and private key for encryption and decryption
* Cost effective