Objective

The objective of this project is to develop and implement an online application system for admission to MSIT program offered jointly by SSN School of Advanced Software Engineering and Carnegie Mellon University. This will help SASE to move from a manual system of distributing, receiving and processing applications to a completely online system.

Current Features

The implemented system will have the following features:

For the applicant:

Register as user: The system shall provide an interface for new applicants to register in order to proceed to the application

Login and logout: The system shall provide a secure authentication system which will permit only registered applicants to manage their applications

Save and resume application: The system shall allow the applicant to save a partially filled application form and resume it later if the applicant wishes to logout and return later

Upload Certificates: The system shall allow the applicant to upload his certificates in df format.

Preview and download: The system shall allow the applicant to preview completed application form and download the previewed application before proceeding to submit

system shall move the application for processing and shall provide him a reciept of payment.

For SSN administrator:

Login and logout: The system shall provide a secure authentication system to the administrator to allow access to the administrator module of the application system

View the applicant details:The administrative assistant gets to view the submitted application of all the applicant. Also if the applicant is evaluated by the professor the professor the corresponding details (suitable,waitlisted or unsuitable) is displayed.

Professor :

Login and logout: The system shall provide a secure authentication system to the professors.

View applicant list : The professors gets to view the pdf copy of the submitted application of the desired appliaction . Also the professors can evaluate each of the applicant based on the applicant and various other criterias as per the current

evaluation form and provide their feedback suitable,waitlist or unsuitable) which gets reflected on the administrative assistant's page.

SYSTEM SETUP:

In the parent folder of SASE website (www.sase.ssn.edu.in) place both SASE app and SASE admin.

Student

Provide a link in the SASE website admissions page which gets redirected to SASE app/registration.html in which new student gets to register.

For the administrative assistant access www.sase.ssn.edu.in/SASE admin/admin_login.php. In order to add a new admistrative assistant use www.sase.ssn.edu.in/SASE admin/admin_registration.php.

For the professors to login , www.sase.ssn.edu.in/SASE admin/professor_login.php.In order to add a new professor use www.sase.ssn.edu.in/SASE admin/admin_login.php.

DB Setup:

Create 2 databases sase_student and sase_admin and import the provided SQL files (schema) into the respective databases. Currently the 'dbcontroller' php files in all the 3 modules, will have 'root' and 'root' as username and password to connect to the DB. This needs to be changed to the correct credentials in the SASE database server to ensure connectivity.

Current Program Details:

SASE app: This module has the below PHP files.

1. Registration.html :The page is used for a new student to login. It taken in firstname,middlename and last name where themandatory fields are marked. Using jQury() function we have perfomed live validation for all the fields.
2. Register.php : On submit from the registration the page gets redirected to this php file. Here the password given the student is salt encrypted and saved in the users table.
3. Login.php: On successful validation register.php page is auto redirected to login.php.here the entered user name is checked with the users table and

password entered is encrypted and the one stored in database is decrypted and compared. The valid user is redirected to the appstatus.php page.

4. The app status page gives the status of the application. If the user is a first time user the status is start application and if he/she has some saved details the status is resume appliaction.
5. The form has tabs for each of pages to to be filled.
6. Personal.php : The appstatus.php gets redirected to instruction page with all the forms as tabs to be filled. Here the user fills all his personal details like gender,dob,address etc.
7. Education.php : Here the user fills the details pertaining to 10^{th},12^{th},UG,PG,gre,toefl .
8. Essays.php: Here a word template is provided for each of the files to be uploaded and photo.
9. Documents.php : This page has upload feature for all the education documents (10^{th},12^{th},UG,PG,gre and toefl)
10. Validation.php : This page provides details of the mandatory fields that are not filled.
11. Pdfpreview.php : This given a preview of all the details filled in a pdf format.

SASE admin:

The files in this module:

1. This has registration,login,logout,profile and change password page for both administrative assistant and professors.
2. Professor_Selection.php : The page displays all the students who have applied along with a link to view the pdf preview of the student details . Another link to evaluate the student is also provided for each student. On clicking on the link the page is redirected to evaluation.php
3. Professor_Evaluation.php : In this form the professor provides evaluation of the students and the corresponding status.
4. Admin_Display.php : This progarm displays the applicant details and the corresponding status provided by the professor to the adminstrative assistant.

Future Work:

1. **Integrating BillDesk:**

2. Please find below the URL where the parameters should be posted to:
3.
4. URL (also referred to as the BillDesk Payment Options page): ========== https://pgi.billdesk.com/pgidsk/PGIMerchantPayment
5.
6. -- The checksum component & checksum password has been emailed in a separate email.
7.

8. -- While this integration (URL) is for a test mode, it is important to note that you can only send Rs 2.00 as the transaction amount. However, you cannot use Re.1 as the txn amount.
9.
10. -- Please note, since the BillDesk payment options page (URL as mentioned above) is secured via your Referral URL, you cannot directly copy/paste the above URL in the browser window to see the page. Only once your 'Referral URL' calls the BillDesk URL would you be able to see the BillDesk payment options page.
11.
12. -- For testing the integration: Once you initiate a transaction (i.e. post the parameters) to the BillDesk URL, on the BillDesk page select 'Visa' and click 'Submit'. On the Credit card gateway page enter card details and complete the transaction, on successful completion of transaction and subsequent browser redirection from the Bank to BillDesk to your Return URL you should receive the return response ('msg' parameter) with the status (and other details) of the transaction.
13.
14. -- During the integration setup at your end, should you face any difficulties, please refer to the BillDesk PG Integration Document -- specifically -- the 'Key Points for a Successful Integration' section and see if you can narrow down on the reason.
15.
16. List of Parameters that will have to be passed onto BillDesk is as follows: - **PLEASE ENSURE THAT YOU WILL NOT PASS ANY SPECIAL CHARACTERS OTHER THEN THE SPACE, @, _ and FULL STOP. IF YOU PASS ANY OTHER CHARACTER THE TRANSACTION MAY FAIL as they are restricted characters and Billdesk will not permit them.**
17.
18.

Parameter Name	Parameter Field Name	Starts With	Minimum length	Max Length	Data Type(Numeric/Alph and special character
txtCustomerID	Txn ID		5	50	Alpha Numeric
txtTxnAmount	Txn Amt		10,2	10,2	Numeric
txtAdditionalInfo1	NA				
txtAdditionalInfo2	NA				
txtAdditionalInfo3	NA				
txtAdditionalInfo4	NA				
txtAdditionalInfo5	NA				
txtAdditionalInfo6	NA				
txtAdditionalInfo7	NA				

19.
20.
21. It is advisable and recommended to use less characters since these are browser based transactions, the maximum character limit is however at 120 characters which is permissible at BillDesk.
22.
23. Please do test at your end along with following the response message hashmatching flow that needs to be implemented at your end as described below:-
24.
25. Once you get the response string back from BillDesk to your return URL, you need to follow the following steps at your end:-

Steps that you need to follow at your end post receipt of response message:-

1) BillDesk will send the response message back to your return url as a "msg" parameter using POST method with a checksum value

2) This checksum value will be the last token in the response message

3) You need to park this BillDesk computed checksum value in a local variable

4) You then need to remove the BillDesk computed checksum value and the preceding pipe (|), thereby this makes the response string as the original response string for which BillDesk has computed the checksum value.

5) You now need to compute the checksum value for this original response string and compare with what was shared by BillDesk (you had parked the BillDesk checksum value in the local variable – refer point no 3 above).

Note:- Please do not break the BillDesk shared response string and re-construct the same to compute the checksum value – by doing this you will be losing the originality of the response string and the chances of checksum value that you computed may not match with what BillDesk computed for the original response string.

6) If both the checksum value matches, then:-

a. Check for **auth status** in the response message -- If the auth status is 0300, then you may update your database for this record as SUCCESS at your end.

b. Check for **auth status** in the response message -- If the auth status is anything other than 0300, then you may update your database for this record as FAILED at your end.

7) If the checksum value does not match, then simply decline the transaction since this looks like a case of someone intercepting the transaction

a. In such cases whatever be the auth status, just decline the transaction and show a failure message.