When an application is deployed on the Web or app store, it simply becomes available to everyone, therefore, developing secure web apps\portal is of the extreme importance specially websites that provide personalized accounts and\or utilize cross platform information, e.g. finance or healthcare web applications.

Some of our best practices and techniques for developing secure web or mobile application are cited below -

URL Encoding - We use mechanism for encoding information in a Uniform Resource Identifier (URI) where needed\possible. Field values are url encoded to avoid security risks.

Input Validation - We devise user data input validation at both Client and Server level (both front end and back-end).

Cryptographic Techniques - We encrypt all confidential data using strong cryptographic techniques where needed\feasible. Credentials encryption at Code level or at DB Level or at both ends.

Avoid CSRF - We create forms in a safe way to avoid cross-site request forgeries (CSRF).

Securing Database Queries - We do this by using prepared statements instead of using string concatenation.

Filter Output Data - Escaping of special characters is very important in order to block certain attacks involving execution of malicious Javascripts by the browsers. We take care of this where feasible.

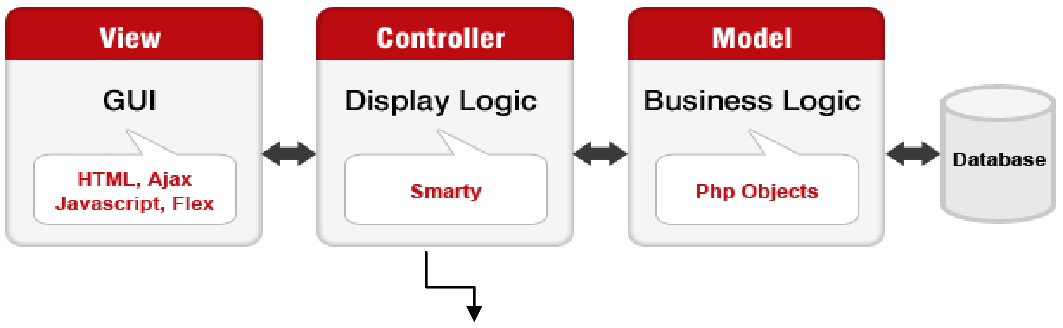
Error Handling - All applications encounter errors when they are being developed and also after they are deployed. The raw error messages returned from application, database, or external programs, should not be displayed to the users directly. The detailed information revealed in error messages might give a clue to malicious users to help them break the application completely. That's why we normally store the error messages in server's error log instead of displaying them to a user.

Session Validations – We enforce user session validations and disable the back button in browsers.

Re-authentication - Where feasible, we try to re-authenticate the users in profiles setting pages and pages involving change in financial accounts credentials information.

SQL Injection Mitigation – A hacker can attack a database by either changing values in the query string that has been determined to be part of an SQL string or by placing the malicious SQL into form fields. We mitigate this using query string manipulation. The query string is a set of values that are passed through the address bar of a browser. It provides an access point for a hacker to the database via a method known as SQL Injection, root access to the machine and access to various parts of the websites code.

Separation of client and server - The user request a page from the client and the server responds by finding that file in its directories and then processes it and finally delivers it back to the user. The server is where all the application's actions are taken place. We use framework methodologies to separate them as illustrated below –



[Or other frameworks e.g Yodlee SDK]

Our best practices covered above are just a few of the mitigation and preventive approach that we take into consideration when develop secure web or mobile apps. Needless to say, these are complimented by server security settings and system security levels. Developing securely written code will help provide adequate defense.