The objective of this document is to provide a set of security checklist for the redesigning of University Website. It provides the security requirements that should be articulated to developers, to make necessary contractual agreement with developers, and to ensure that the necessary testing is completed prior to deploying these applications.

The security requirement below is intended to provide a basis for securing University websites, web applications, and databases from malicious and unintentional abuse.

- **Risk Assessment**
  All applications should be examined to determine if it is connecting to a critical database and the confidentiality, criticality, and vulnerability assessed. The mitigation of risk to a realistic and acceptable level is required for all applications that will be connected to critical databases. The Family Educational Rights and Privacy Act (FERPA) mandate that the confidentiality of student data be protected. Databases that store personally identifiable information (PII) requires appropriate security in place to protect the confidentiality of this data.

- **Authentication**
  This is the first line of defense in protecting critical data asset. To ensure that the application denied unauthorized access to critical data, there should be a designed, developed, and documented authentication polices, processes, and logging that will identify the true owner of a user id and password. The policies and processes should enforce strong password with a minimum of eight characters with a mixture of numbers, upper case letters, lower case letters, and symbols.

- **Authorization and Access Control**
  This provides a security strategy to protect the front-end and back-end data, operating system, and modification of applications. Deploying a role base approach where users are defined based on a least privilege model. There should be a description of all the fields and tables, the expected values of a field and data length, and permissions assigned to fields.

  Ensure that users cannot browse beyond their user role rights. Prevent user activity from being cached when handling sensitive data. Penetration test should be conducted to assure that all access control has been tested and denied unauthorized access prior to deploying the application.
• **Session Management**
  Deploy security measure that will protect the session IDs throughout their life cycle. This will prevent an unauthorized user from hijacking a session and assume the valid user’s identity.

• **Data and Input Validation**
  Address Cross-Site Scripting, command injection, buffer overflows, and error handling during the design and development phase of the application. Prevent the applications from creating an output from user input without validating the data. Limit user access to only SQL stored procedures that are required. All applications that accept input from users via a HTTP request must be reviewed to ensure that it can identify large input. All inappropriate data that is identified should be dropped and activity logged.

• **Web Application and Server Configuration**
  Patch management policy and procedure should be implemented that will ensure that “out of the box” server/application vulnerabilities are addressed. Implement server deployment policy and procedure that will ensure that only processes that are needed turned on and unused port closed.

• **Logging and Auditing**
  Implement logging and auditing policy that will provide individual accountability and is a requirement in any legal proceedings.