

Web Development

About This Policy

Responsible Office

Management Information System

Legal Basis

CSU Code Book IV Chapter 1 Article 96 Section 1

1. Purpose

Caraga State University's Management Information System (MIS) has

The purpose of this document is to define the components of the University's supported web development platform, coding standards, and testing and approval process so that applications can be developed in a manner consistent with accepted interoperability and security practices and be fully compatible and supportable in our environment.

2. Scope

Any University division, department, or individual that develops applications that will run on MIS or co-located platforms. This includes both web-based and traditional client/server based applications.

3. Acceptable Technologies

In general, in-house developers or not affiliated in MIS unit hired to develop custom web-based applications for use on MIS supported servers should develop those applications using **open-source** software or projects, languages, and tools. MIS defines open-source to mean:

"A software or project whose source code is freely available for anyone to view, modify and distribute."

The list of acceptable open-source technologies that are supported by MIS include but are not limited to:

- JavaScript / ECMAScript. MIS prefer the following technologies:
 - Vuetify for User Interface
 - NuxtJS for Front-End application
 - NestJS for Back-end API

- Yarn Package Manager
 - PM2 for deployment
- SSL/TLS
- Nginx
- RDBMS (SQL Lite, PostgreSQL, MySQL, Firebird)
- Lightweight **D**irectory **A**ccess **P**rotocol (Open LDAP)

Note: Web developers intending to use any technologies other than those listed above must consult with MIS before any development work begins. MIS cannot guarantee application compatibility with our existing infrastructure if non-supported technologies are used for development or deployment.

4. Platform and Functionality Considerations

To ensure application compatibility within CSU's campus computing infrastructure, web application developers should keep the following in mind:

- Production web applications servers at CSU primarily use NginX 1.x on Linux.
- While MIS primarily uses NodeJS, all JavaScript web application code must be written with node long term version (LTS) with a minimum of 20.x and yarn version 1.22.x and above.
- It is highly discouraged to use third-party databases with subscription-based functionality. However, the preferred database platforms for web-based applications are Postgres, MySQL, and SQL Lite.
- Web-based applications must support recent versions of all popular web browsers, including Mozilla Firefox 17 or higher, Microsoft Edge or higher, and Safari 5 or higher. Adhering to W3C web standards is the best way to ensure this compatibility.
- Any user authentication mechanisms must provide an encrypted (SSL) HTTPS connection for the login screen to avoid transmitting username and password information in plain text. MIS can provide SSL certificates upon request.
- Authentication mechanisms may use the campus LDAP server.
- Any file transfer operations, SQL queries, or directory service lookups must occur over a secure channel such as SSL, SFTP, or SCP.

5. Application Verification Testing and Development Lifecycle

- Applications should be designed based on the platforms, tools, and data connectivity guidelines presented in this document and other related University policy documents such as Data Privacy Manual.
- Functional requirements for applications should consider all appropriate University policies, and data privacy for secure access, handling of sensitive data, and protection of personally identifiable information (PII) or financial records.
- Whenever possible, application development will be performed in a secure 'dev' or 'test' environment that is isolated from the Internet and may have limited or no access to the University's production server farm and campus network.
- Prior to moving an application from the dev/test environment to production use, the application will be scanned by MIS's Systems and Security Group for known security vulnerabilities using automated tools such as WebInspect, AppScan, or other open-source utilities. Application developers are encouraged to request periodic security scans during the development process (i.e. at each milestone of the project) to pro-actively address security vulnerabilities and reduce the likelihood of issues arising during the final pre-production scan.
- When the pre-production application scanner has been completed, the application will be moved into the appropriate production environment and any required external firewall rules for remote communication will be enabled.
- MIS's Systems and Security Group will periodically re-scan applications that are in production use to ensure that they are not vulnerable to new attack methods.
- For other web applications not compatible with the prescribed technology or framework, the developer is the sole responsible for whatever risk may occur.