EXHIBIT A

Acceptance Criteria:

1. Drupal 7 to Drupal 11 Platform Upgrade

- **Current State Assessment:**
  - Conduct a thorough assessment of the existing Drupal 7 site to identify all components, modules, layouts, features, custom code and functionality.

- **Upgrade Plan:**
  - Develop a detailed upgrade plan outlining the steps for migrating from Drupal 7 to Drupal 11, including timelines and key milestones.
  - Plan should at-least include all items identified in the scope & acceptance criteria.

- **Compatibility Review:**
  - Review and update all modules and themes to ensure compatibility with Drupal 11. Identifying improvements or replacements where necessary.

- **Open-Source Commitment:**
  - All contributions or patches on issues identified during this project related to improving Drupal core, established modules and themes are required to be contributed back to the Drupal Open-Source community.
  - Contributions outside of Drupal's ecosystem is similarly encouraged.
  - These contributions should be identified for the CWI Web Team so they can participate in continued development, support, and testing to nurture commits on those improvements.

- **Testing:**
  - Perform extensive testing to ensure all features and functionalities work correctly in the Drupal 11 environment.
- **User Acceptance Testing (UAT):**
  - Facilitate UAT sessions with key stakeholders to validate the upgraded platform meets all requirements.

- **Documentation:**
  - Provide comprehensive documentation of the new Drupal 11 environment, including any custom code changes and configurations.

2. Database Migration

- **Data Inventory:**
  - Conduct a complete inventory of the current database, including all content types, fields, URL, users, and configurations.

- **Migration Strategy:**

- Develop a detailed migration strategy to transfer all published data from Drupal 7 to Drupal 11.
- Improve any identified content entities and fields to align with Drupal 11 best practices for performance and management.
- Final data migration with deployment of the production environment should be performed in coordination CWI Web Team. To include steps to hold Drupal 7 content changes, incorporate any manual webpage changes, and validate production webpages and functionality.

- **Data Integrity:**
  - Ensure data integrity is maintained throughout the migration process, with no loss of data.
  - Identify data/webpages that may need manual attention (like text areas with PHP, CSS, JavaScript) following a migration to ensure style guides or external embedded elements remain functional.

- **Revisions:**
  - Build strategy to maintain old revisions for existing published content, knowing that some may need to be dropped in the new platform based on the established content entity strategy.

- **URL Continuity:**
  - Ensure current URL usage is maintained across the site to include URL's that are: current published content, previous content that has been redirected via alias names and node id's, IMCE files etc.

- **Performance Testing:**
  - Conduct performance testing post-migration to ensure the new platform handles the current load efficiently.

3. Establish Digital Style Guide with Drupal Component Integration

- **Pattern Library:**
  - Establish a Pattern Library in Figma, using atomic design principles
  - Create a comprehensive pattern library that outlines brand colors, typography, iconography, and other visual elements.
  - Align with digital style guide variables to synchronize styles for consistent UI Kit Management and future prototyping with established variables.
  - Include newly established brand changes like switching to the variable font "Archivo" and align with new button styles.

- **Digital Style Guide Development:**
  - Utilize Storybook (with Twig) to establish a digital style guide.

- Include Storybook controls, add-on designs to Figma, align with responsive breakpoints, sample pages to related to our Drupal page layouts and content type options.
- Translate established visual components and JavaScript functionality into the digital style guide.
- Incorporate newer component styles and/or functionality changes - established with existing UX vendor Collegis established by wireframes
- Following industry best practices to recommend improvements where needed, typically identified during development stages of the style guide.
- Establish a workflow to export the digital style guide's Storybook website to a public facing location.

- **Component Library:**
  - Develop a reusable component library within Drupal 11 that adheres to the established digital style guide.
  - Apply the same atomic design structure established with the pattern library and digital style guide development.
  - Follow the Single Directory Component concepts.

- **Integration:**
  - Integrate the pattern library & digital style guide components into Drupal 11 components and theming code, ensuring consistency across Drupal and the design system.
  - Reduce duplications in development workflows of new Drupal 11 components and digital style guide evolutions.
  - Utilize a supported open-source concept to provide this integration. (e.g., Emulsify).

- **Responsive Design:**
  - Ensure all components and page layouts are fully responsive
  - Relative to browser font size
  - Optimized for various devices and screen sizes (e.g., iPhone 6 to 1800px+ @ 1x and 2x pixel densities).

- **Accessibility Compliance:**
  - Ensure all design components comply with WCAG 2.1 **AA** accessibility standards.
  - Strive for WCAG 2.1 **AAA** accessibility standards with all components that can reasonably reach it.

4. Site Improvements beyond Current State

- **Content Calendar Dashboard:**
    - Process where an editorial role can manually setup a notification-based entity, connecting a due date and page that needs edited.
    - Reoccurring due dates can also be established (e.g., month, semester, year).
    - These notifications can be connected to all relevant entities (e.g., Page, File, Program, Policies).
    - Email notifications are sent page authors and editorial roles at intervals prior to due date.
    - Administrative based dashboard to provide relevant reports and manage notifications.
    - Age based reports outside or manually set notifications should also be available, based on last updated date.

- **Content Security Policy (CSP) Ready:**
    - Inline JavaScript and CSS styles should all be removed preparing for application of a CSP that does not require decelerations for: 'unsafe-inline or 'unsafe-eval'.
    - Anything that cannot reasonably be resolved must be identified with rational to plan a path to remediate.

- **Newsletter Design:**
    - Webpage and Email designs improved, based on wireframes.

5. Establish Hosting Infrastructure and Web Operations Processes

- **Version Control with GitHub:**
    - Connect with a GitHub repository for version control, ensuring all code changes are tracked and managed efficiently.

- **Local Development Environment:**
    - Establish a standardized local development environment for developers, including documentation on setup and usage, tooling and configurations stored in code. (e.g., Lando).

- **Continuous Integration (CI):**
    - Implement a CI process to automate testing and deployment, ensuring code quality and reducing manual intervention. (e.g., Circle CI).

- **Pantheon Multi-Dev Environment:**
    - Utilize the Pantheon multi-dev environment and WebOps workflows to support parallel development streams, allowing for efficient collaboration and testing across our teams.

- **Pantheon Production Environment:**
    - Set up and configure the production environment on Pantheon, ensuring it is optimized for performance and security.

- Establish process for Production environment's access and error logs to integrate with CWI enterprise security tools like SIEM (Security information and event management), IDS/IPS (Intrusion Detection and Prevention Systems) and vulnerability scanners.

- **WebOps Workflow:**
  - Build a WebOps workflow to streamline local development changes, automate the design system needs, build processes w/ unit testing, and execute deployment processes.
  - Ensure a continuous improvement methodology to provide consistency and reliability across environments.

- **Documentation and Training:**
  - Provide documentation on the hosting infrastructure and WebOps processes, to include tooling and necessary training and initial support for the CWI Web Team.

6. Additional Services

- **SEO and Analytics:**
  - Implement SEO best practices on page (e.g., Metatags, JSON-LD) and integrate into our analytics tools (e.g., Google Analytics & Google Tags) to track and report site performance.
  - Transition & improve current SEO fields and metatags configurations.

- **Core Web Vitals:**
  - Ensure Google Core Web vitals testing reaches "Good" statuses for elements that relate to the Drupal 11 hosted environment.

- **Security:**
  - Implement security best practices and ensure the Drupal 11 site adheres to relevant industry security standards and regulations (e.g., HIPAA, PCI-DSS, GDPR).
  - Must have robust security features to mitigate cyber threats, including:
    - Secure authentication and authorization mechanisms
    - Regular security updates and patches process
    - Protection against common web vulnerabilities (XSS, CSRF, SQL injection, etc.)
    - Encryption of data at rest and in transit
    - Audit logging and monitoring capabilities

- **Training and Support:**
  - Establish education-based tasks, research topics and resources for the CWI Web Team to prepare them for continued innovation after project deployment. Can include Drupal 11 configuration, module, component and theme development, Style Guide expansion, etc...

- Develop training sessions for web content contributors, editors and administrators on the new Drupal 11 platform focused on their roles.
- Offer post-launch support for a minimum period of 90 days to address any issues and provide developer assistance as needed.

- **Long Term Support Options:**
  - This should not be included in the final quote price for this Migration project, but as future options to consider after project completion for a longer-term partnership and budgeting needs.
  - Please include itemized support options with current pricing that can cover Drupal core, module & security updates, troubleshooting, user experience, content writing, and search engine optimization tasks related to future support, project help or team augmentation.

7. Project Management and Communication

- **Project Plan:**
  - Provide a detailed project plan with clear timelines, milestones, and deliverables.

- **Execution:**
  - An Agile style approach to project execution is encouraged
  - Include the CWI Web Team in planning and execution for tasks when required.

- **Regular Updates:**
  - Schedule regular progress updates (e.g., Sprint Reviews) and any additional meetings with key stakeholders.
  - Prepare a President & All VP's (President's Circle) focused update to share regularly (e.g., each month).

- **Issue Resolution:**
  - Establish a process for tracking and resolving any issues that arise during the project.

8. Acceptance Testing and Go-Live

- **Final Testing:**
  - Conduct a final round of testing, including load testing and security testing, before the go-live date.

- **Stakeholder Sign-Off:**

    - Obtain sign-off from all key stakeholders before the final deployment.

- **Deployment:**
    - Deploy the upgraded site to the live environment with minimal downtime for anonymous visitors.
    - Downtime for website contributors is expected but should be well communicated and minimized to ensure the final database migration and any manual page building identified during project phases can be completed prior to full deployments.

- **Post-Launch Review:**
    - Conduct a post-launch review to ensure everything is functioning as expected and address any immediate issues.

Scope:

- **Focused on CWI.edu a single Drupal 7 platform**

- **Migration of website features to a Drupal 11 platform – Content, Design, and Functionality**

- **Build an atomic design pattern and style guide from existing site style and features**

- **Integrate Style guide and Drupal 10 Component library for alignment and reduce development duplication**

- **Establish WebOps hosting model with developer toolsets, incorporating security best practices**

- **Organization Dependencies:**
    - SSO – login - ADFS
    - My.CWI.edu SharePoint and Ellucian Experience platforms – News XML feeds
    - Careers.cwi.edu - Site Template and Custom CSS/JS - PageUp
    - Search.cwi.edu - Site Template, XML Feeds provided, Search Indexing On/Off in page templates - Funnelback
    - Colleague / SQL – Faculty and Staff Directory feed – CSV ingested with Feeds and custom node processing scripts

- **Third Party Dependencies:**
    - Content Security Policy
        - Report Mode currently
    - Google GA4 – Changes coordinated with Collegis implementation

- Tag Manager – 1st Party Cookie setup for forms, Digital Marketing Ad's and Tool integrations
- Microsoft Clarity – Session Recording
- VWO Visual Website Optimizer – A/B Testing
- Element 451 – Tracking and Chatbot Pixel sitewide
  - Sitemap important for Chatbot Knowledgebase
- SiteImprove – Accessibility Checking / Content Quality
- Ads Pixels – Google Conversion and Remarketing, Facebook, Linked-in, HubSpot, KTVB
- Ocelot Chatbot – Targeted to Financial Aid pages, GA4 Tracking
- Resmush.it - Image file size optimizer – in many image styles

- **Drupal 7 Modules - Noteworthy:**
  - Redirect / Global Redirect – May need some custom mappings to fully drop old nodes
  - Pathauto - to manage URL aliases – need to transition
  - XML Sitemap / Metatags / Structured Data
  - Revision Management with Workbench Moderation
    - Past Revisions can be dropped during migration
    - But a Moderation Workflow is important to maintain
  - Workbench Access used to control user access to individual nodes
  - SSO with ADFS – using MiniOrange Module
  - Google Analytics Reports module - showing 30- & 90-day charts on page for contributors – Broken July 2024, Needs realigned with GA4
  - Automatic Entity Labels – Some content types
  - Book – important our Policies content type
  - Boxes – Exported into Features
  - Classy Panel Styles – Few background options used for Panels and Panelizer layouts
  - Colorbox – images primarily, might be Colorbox node used on a few pages in CKeditor
  - Date – important for a few content types
  - Diff – Content Moderators use heavily
  - Entity Reference – heavily used across content types and views
  - Entityforms
    - Give form – will be removed
    - Facility / Event Request – will need a solution – field heavy and low submissions
  - Newsletter Subscribe – just captures email addresses - probably a better solution
  - EVA – attaching views to node display
  - Font Awesome w/ Icon Picker for Button Components
  - Feeds – Sync with CSV file for Faculty and Staff / Person nodes
  - Mandrill – System and Newsletter configurations
  - Simplenews Newsletters with subscribe and list management for 1 external list

- Mobile Codes Module – Automated QR code images on News and Programs
- Search 404 – Redirect to Funnelback Search
- Search API w/ Facets – Database storage used with Policies & News Search
- Views based Search for Faculty and Staff and Student Forms
- Cron – Scheduled with some PHP cleanup scripts
    - New Relic pings cron every 5 mins
- Custom tokens for Metatags – Pulling relevant images for Social off the page
- Field Redirection - used to hide some content type node display or with some Programs that redirect to other nodes but display in search
- FlexSlider – Slideshows like TV Announcements
- Image Styles – Extensive usage, some shared across fields
- Picture and Breakpoints – To show right sized image styles for devices and lazy load images for performance, across all content types
- Manual Crop – to provide contributor-based cropping of images across content-types, primary image styles with scaled down dependencies
- IMCE – Needs Cleaned Up and removed, but some files still are in use, have been building external FTP for new files
- Job Scheduler – used to trigger publishing of certain nodes – TV Announcement, Occasions and Hero's
- Modernizer – for some CSS selectors
- Page Manager – External Page templates, Newsletter Archive, FAQ's
- Panels – Extensive used for Node display, some multiple variants based on taxonomies, 9 layouts
- Panelizer – Extensive Node controls for Page primarily and Webform content types
- Paragraphs – Extensive usage for Page Content Type, new format we have focused on with Layout, Component and Style Control based elements
- PDF – viewer to show PDFs on File content type page
    - Concerns with Old PDFs in revisions found by Google
- Rest Server – Providing Funnelback xml for customized search's – Programs and Faculty and Staff Directory, Program Finder
- Rules – Notifications and some node management help
- Structured data JSON-LD – not implemented well, but would like to expand
- Taxonomy Term Permissions – Used to control editing/access to some tags
- Themekey – TV announcements switch to Zen_CWI theme
- Views PHP – will need to transition out, but a few views do exits
- Views Content Cache – helpful on frontend performance
- Webforms with Clientside Validation
    - Many webforms are transitioning to Element451, but a few will remain.

- **Content Types with Short Description of Usage:**
    - Announcement
        - TV image slideshow displayed at various locations around campus
        - Taxonomy controlled display categories

        - Taxonomy URL is supplied to TV's just shows a slideshow of images that refresh every 5 mins
        - Using Brightsign devices on TVs to Open URL's and reboot daily
        - Scheduled publishing
        - Secondary Theme (Zen) used which is just blank except for the Flex Slider View
        - Admin Console built
    - Blog
        - Dismantle
        - Transitioned to News as was a duplicated structure
    - Club
        - Student Clubs
        - Singular Web Contributor
        - References Social and Person CT
        - Referenced by News and Programs
        - Panels and views layout
        - Views based Search - https://cwi.edu/campus-life/student-clubs
    - Contact Us
        - Dismantle
    - Department
        - Referenced by Program
        - More of just a taxonomy at this point
        - no individual node display
    - FAQ
        - Q / A & Taxonomy
        - Panels and views layout
        - View based Search and Category Page - https://cwi.edu/faq
        - No longer in navigation, but still trying just have other nodes reference it
        - Referenced on Pages & Programs
    - File
        - Pages with embedded forms or pdf
        - Related to enrollment, current students or program-based documents
        - PDF viewer
        - File force downloads
        - Panels and views layout
        - Views based Search - https://cwi.edu/academic-advising/student-forms
    - Hero
        - 3 Main Display Types
            - Video with Mobile / latency image fallback, play button, text overlay
            - Image with text overlay – mostly used
            - Text – Emergencies and Alerts
        - 3 Levels - Placement
            - Emergency – Sitewide
            - Alert – Homepage, Main Support and Digital entry pages only

        - Homepage – Homepage only
    - Scheduled publishing
    - Block based displays
    - Views PHP used
    - Admin Console built
    - Requires cache clearing in Emergency situations
- Homepage Slide Collection
    - Dismantle
- Location
    - Address, Services, Picture, Map
    - Views based layout currently
    - Working on transition Paragraphs layout for more flexible page builds
    - no individual node display – but have ideas to enable this
- News
    - Panels and views layout
    - Multiple Taxonomies and control fields in use
    - Views for various internal page components and external feeds
    - Blogs migrated recently – will need new standard layout for blogs articles and new views built
    - Referenced in newsletters
    - Refencing Person, Clubs & Programs
    - Model to unpublish news at certain dates
    - More Flexible CKeditor style implementation desired
    - Search API based News Search with filters - https://cwi.edu/news
    - Related News views
    - Admin Console built
- Newsletter – Communications
    - Will need redesigned
    - Page and Email layouts
    - Field Collection Heavy
    - References Spotlights & News CT's
    - 2 Newsletters
        - Internal – Weekly Send
        - External – Monthly send
    - SimpleNews module Implementation
    - List Management for External Newsletter with automated sending
    - Newsletter Archives should be maintained
    - New Designs for Page and Email
        - Ready to share
        - Requires field and functionality changes
- Occasion
    - Used in Paragraphs component
    - no individual node display
    - Scheduled publishing
    - Admin Console built

- Page
  - Panels / Panelizer based layout controlled per node
  - Lots of Web contributors' editing, have been reducing them over the years
  - 2 main layouts
    - Panelizer
      - Few layouts to choose from with various regions
      - Many Content Panels views for contributor to pick from
      - Lots of related fields some shared with multiple components
      - Field Collection Heavy
      - Custom JS built for many content panels
    - Paragraphs
      - 1 Panelizer layout
      - Paragraphs editor-based editing
      - Layout and Component based paragraphs
      - Views based paragraphs
      - Custom JS built for many components
      - Style control-based fields with module code to control CSS classes
      - Background color and image implementations for layout paragraphs
- Person
  - Feeds managed content and publishing status
  - Panels and views layout
  - Not editable by contributors
  - Some manual Person records – like department contacts and board members
  - Node display – incorporates Person Bio fields
  - Daily Feeds synchronization by hash from an IT managed CSV file out of Ellucian Colleague
- Person Bio
  - Referenced by Person
  - Contributor Editable – Bio, Headshot, Location
  - no individual node display
  - Daily PHP script to manage publishing based on related Person nodes
- Policies and Handbooks
  - Book Module with Print capabilities
  - Panels and views layout – unique to each book
  - Search API based Search component
  - This is expanding with new books regularly
  - Structured Data could use refinement but working
- Program
  - Main Program content
  - References to other content types

        - Various control and filtering options
        - PHP Template for display and logic
        - Include some old Hidden fields from a recent redesign and related views we can dismantle
        - Admin Console built
    - Program – Academic Date
        - Dismantle
    - Program – Degree
        - Referenced by Program
        - no individual node display
        - Degrees matched to catalog
        - Automated PHP script to update degree costs annually from CSV
        - Interval field – logic to switch between months and years when appropriate
    - Program – IDOL
        - Referenced by Program
        - no individual node display
        - Entity is a CIP code for related Programs
        - Paragraph entity for Occupation
            - Career, Description, Openings, Wage, and Growth data
        - Should be annual updates or when mapping change
        - Automated PHP script to update annually from CSV
    - Program – NC-SARA
        - Professional Licensure transfer data required for Compliance
        - Referenced by Program
        - Paragraph entity for each State
            - Decisions, Description, Link
        - Individual node display - https://cwi.edu/ncsara/licensed-practical-nurse-lpn
        - About 5 programs use currently, expected to expand
    - Social
        - Collects Social Account for Clubs
        - Referenced by Clubs
        - no individual node displays
        - Did have a search page associated, no longer in use
    - Spotlight
        - Capture Image and Description for Primary Newsletter section
        - Referenced by more than one Newsletters
        - no individual node displays
    - Testimonial
        - Image, Quote, Name, News Article Reference
        - Referenced by Program's
        - working on Paragraph display for Page's like Homepage
        - no individual node displays
        - Admin Console built

- Webform
  - Forms
  - Many Transitioning to Element451 over the next few months
  - Panelized but rarely extra content