

# **REQUEST FOR PROPOSALS**

**For:**

**myCITC Phase 2**



**Cook Inlet Tribal Council, Inc.**  
**August 28, 2023**

## Contents

myCITC Phase 2 - RFP .....	3
Summary .....	3
Background .....	3
Questions .....	3
Contract Term .....	4
Timeline .....	4
Budget .....	4
Project Goals .....	4
Requirements .....	4
Scope of Work (SOW) .....	5
General Conditions To Proposers .....	8
RFP Conditions and Provisions .....	8
RFP Response Requirements .....	10
Evaluation Process .....	11
Scoring Criterion .....	12
Award .....	13
Insurance Coverage .....	13
Restrictions on Lobbying Activities .....	14
Appendix A: Current Solution Specifications .....	16

# myCITC Phase 2 - RFP

## Summary

Cook Inlet Tribal Council, Inc. (CITC) is looking for a design and build software development firm to support the ongoing development and maintenance of its myCITC Participant Portal.

CITC seeks proposals from qualified software development vendors to continue enhancement and expansion of the existing CITC Participant Portal. This portal hosted in CITC's Azure environment aims to provide CITC staff and program participants with tools to connect, access services, and manage profiles online.

**Proposals must be submitted by email (Hard copies will not be accepted).** Proposals should be sent to Andrew Chlup at [ALChlup@citci.org](mailto:ALChlup@citci.org) no later than 12:00 PM AST Sept 25th, 2023. Proposals received after that time and date will be rejected.

## Important Dates

<b>August 30<sup>th</sup>, 2023</b>	<b>RFP Posted</b>
<b>September 15<sup>th</sup>, 2023</b>	<b>Questions Close</b>
<b>September 25<sup>th</sup>, 2023</b>	<b>Proposals Due</b>

## Background

CITC is a tribal non-profit organization serving Alaska Native and American Indian people living in Southcentral Alaska. CITC helps individuals achieve their full potential through an array of support services including youth education, employment and training services, workforce development, addiction and recovery, and other programs supporting healthy families. For more information refer to [www.citci.org](http://www.citci.org).

When applicable, CITC requires a Native-hire preference in all projects and activities per its authority under PL 93-638. To the greatest extent feasible, CITC will give preference in entering into contracts and similar agreements to Alaska Native and American Indian tribes, organizations and economic enterprises. Positive efforts should be taken to utilize small businesses, minority-owned firms, and women's business enterprises whenever possible.

## Questions

CITC shall not be bound by any oral interpretation of this RFP. Questions are encouraged, and should be sent to Andrew Chlup by email at [ALChlup@citci.org](mailto:ALChlup@citci.org) no later than 5:00 pm September 15th, 2023. Substantive issues will be addressed by addendum.

No communication is to be directed to any other CITC employees or CITC representatives.

## **Contract Term**

The awarded contract will be in effect for fiscal year 2024 (Oct 2023 - Sept 2024).

## **Timeline**

This contract is for a 1-year term. Individual project timelines will be defined in each SOW.

In general, software features will be rolled at least quarterly, and any bug fix / small enhancements being released weekly.

## **Budget**

Total budget for the 1-year term shall not exceed \$1,000,000. Individual project budgets and spending authority will be defined in each SOW.

## **Project Goals**

- Refine and optimize existing portal modules including Service Directory, User Profiles, and Secure Messaging
- Develop new modules for appointment scheduling/bookings, targeted alerts based on profile data, and data integration from portal back into CITC systems
- Ensure portal allows self-service access to resources and frictionless experience for diverse users
- Expand features and functionality incrementally over the next year via agile development sprints and new SOWs

## **Requirements**

- Must utilize existing portal technology stack and tools. Detailed information can be found in Appendix A: Current Solution Specifications
- Willingness to work in an agile development methodology
- Provide ongoing maintenance and refinement of existing deployment pipelines into CITC's Azure environment
- Ensure compliance with all applicable regulations including HIPAA
- Adhere to state and federal data privacy laws
- Support accessibility standards and security protocols around data transmission and storage

## Scope of Work (SOW)

This will be an open design/build contract with specific projects defined in SOWs.

The selected vendor will be awarded specific projects to be completed as part of the overall RFP. Projects may include design, development, testing, documentation, training materials, and deployment into CITC's Azure tenant.

Specific deliverables will be defined in each project SOW.

### Example SOW Deliverables

Typical deliverables may include:

#### UI/UX Design

- User flows - Visual diagrams showing the step-by-step paths users take to complete key tasks.
- Wireframes - Black and white schematics visualizing page layouts, content, features, and information architecture.
- Interactive prototypes - Clickable prototypes to demonstrate UI behavior and interactions.
- Visual mockups - Full color visual designs showing spacing, typography, imagery, etc.
- Style guide - Reference for design patterns, branding elements, code snippets, and assets.
- Usability report - Findings and recommendations from usability testing sessions.
- Accessibility checklist - Review of how design meets accessibility standards and guidelines.
- Assets - Final files needed for development like image files, icon sheets, font files, logo files.
- Documentation - Annotations detailing interactions, transitions, responsive behavior, etc.

#### System Architecture

- Architecture diagrams - Visually illustrate the overall system design, including components, connectivity, infrastructure.
- Data models - Details on the database schema, entities, attributes, relationships.
- API specifications - Technical specs for any APIs to be developed or integrated with.
- Integration plan - Overview of how the system will interface with other tools/platforms.
- Infrastructure description - Details on server setup, hardware requirements, network topology.
- Hosting plan - Recommendations on hosting provider, scaling requirements, availability, costs.

- Security overview - Summary of encryption, access controls, and other security measures.
- Scalability plan - Strategy for handling user/traffic growth and scaling up components.
- Performance benchmarks - Metrics like response times, throughput, and capacity for load testing.
- Technology stack - List of specific languages, frameworks, tools, and platforms to be used.
- Testing/QA plan - Summary of testing techniques, automation, and tools to ensure quality.

## Front-end and Back-end Programming

### Frontend:

- Code - HTML, CSS, JavaScript source code files.
- Frameworks - Details on frameworks like React, Vue, Angular, etc.
- Browser support - List of target browser versions for compatibility.
- Responsive design - How UI adapts for mobile, tablet, desktop.
- Accessibility - Compliance with WCAG standards.
- UI components - Reusable UI components, widgets, buttons, forms.
- Testing - Automated tests (unit, integration) and coverage reports.
- Comments - Code documentation and developer comments.
- Assets - Image, font, audio, video files.

### Backend:

- Code - Source code for server-side application logic.
- Database schema - Schema definition for database model and entities.
- APIs - REST API documentation and endpoint details.
- Integration - With external services, data sources, third-party APIs.
- Testing - Automated API tests, unit tests, coverage reports.
- Deployment - Scripts, config for deployment on servers.
- Monitoring - Performance monitoring, logs, alerting.
- Security - Encryption, access controls, authentication, authorization.
- Scaling plan - For handling increasing traffic/load.
- Comments - Documentation of code and technical concepts.

## Reporting and Analytics

- Requirements - Description of key reports, dashboards, metrics needed.
- Data sources - Details on where data will be pulled from.
- Data models - ER diagrams of analytics data structures.
- ETL processes - Extract, transform, load processes for preparing data.
- Visual mockups - Sample visual designs for dashboards, charts, graphs.

- Visualizations - Interactive charts, graphs, gauges, maps etc.
- Dashboards - Interactive dashboards with filters, parameters, etc.
- Reports - Static report templates with layouts, formatting, styles.
- Queries - Actual queries, database views, stored procedures used.
- Alerts/Notifications - Definitions of alert rules and notification triggers.
- Access controls - Permissions for reports, dashboards and data.
- Caching - Approach for optimizing performance of queries/reports.
- Automation - Details on scheduled tasks, refreshes, emailing.
- Code - Any custom programming logic involved.
- Testing - Test cases for functionality, security, performance.

## Documentation and Training materials

- User documentation - User guides, manuals for end user functionality.
- Technical documentation - Dev guides, API docs, architecture overviews, infrastructure docs.
- Admin documentation - Guides for sysadmin tasks, configuration, maintenance.
- Training content - Presentation decks, videos, interactive tutorials.
- Quick start guides - Short guides to get started quickly.
- FAQs - Frequently asked questions and answers.
- Style guide - Documentation style conventions and templates.
- Screenshots - Images used in documentation for visual aid.
- Cheat sheets - Quick reference guides for frequent tasks.
- Support docs - Help desk knowledge base articles for support.
- Changelog - Release notes on updates, bug fixes, known issues.

## Software Testing and QA

- Test plans - Outline scope, approach, timeline, environments for testing.
- Test cases - Detailed test cases specifying steps, inputs, expected results.
- Test data - Test data to be used for functional, integration, system testing.
- Automated tests - Unit tests, integration tests, regression tests in code.
- Testing reports - Results from executed test plans and cases.
- Bug reports - Reports detailing bugs found, steps to reproduce, severity.
- Traceability matrix - Map of test cases to requirements.
- Coverage reports - Code coverage percentage from automated tests.
- Defect tracking - Use of issue tracker like JIRA to track bugs, fixes.
- Performance tests - Load, capacity and stress testing and results.
- Security tests - Penetration, vulnerability test results and fixes.
- Accessibility tests - Compliance with accessibility standards.
- Test environments - Details on tools, software, data used for testing.

## **General Conditions To Proposers**

The general rules and conditions which follow apply to all proposals issued by CITC unless otherwise specified.

This RFP does not obligate CITC or the successful Proposer until a written contract is approved and signed by both parties. When the selection process is complete, successful agency will be expected to agree to the terms and conditions and enter into a contract with CITC memorializing the services to be performed.

**Completeness/Authorization of Proposal:** Proposer shall supply all information and submittals required by the proposal documents to constitute a proper proposal. The proposal shall clearly state the legal name, address, telephone number, and fax number of the proposer. The proposal shall be signed above the typed or printed name and title of the signer. The signer shall have the legal authority to bind the proposer to the proposal.

**Collusive Proposing:** The Proposer certifies that the proposal is made without any previous understanding, agreement or connection with any person, firm, or corporation making a proposal for the same project, without prior knowledge of competitive prices, and that the proposal is in all respects fair, without outside control, collusion, fraud or otherwise illegal action.

**Subletting of Contract:** Proposer shall not assign, transfer, convey, sublet or otherwise dispose of the contract or their right, title or interest therein, or their power to execute such contract to any other person, firm or corporation without the prior written consent of CITC, but in no case shall such consent relieve the Proposer from their obligations, or change the terms of the contract.

**Conflicts of Interest:** Proposer affirms that to the best of Proposer's knowledge there exists no actual or potential conflict between Proposer's family, business, or financial interests and the services to be provided to CITC under a contract with CITC, and in the event of change in Proposer's private interests or services under a contract, Proposer will inform CITC regarding possible conflict of interest which may arise as a result of such change

## **RFP Conditions and Provisions**

- Proposers are advised that the information presented in this document is preliminary. The proposed program, scope, schedule, etc. are subject to refinement and change.
- CITC is not liable for any costs incurred by Proposers prior to the issuance and execution of a contract with the successful agency as a result of this RFP process. All proposal preparation and other costs in responding to this RFP is the sole responsibility of the Proposers.



- CITC reserves the right to waive any informalities, and to make a selection as deemed in its own best interest. This includes the right to reject any or all proposals and the right to proceed utilizing a different process.
- CITC may require, seek, and utilize all information it deems appropriate in order to assess the qualifications of individual Proposers. All information in proposals submitted in response to this RFP is considered confidential and will not be released to the public.
- In the event it becomes necessary to revise any part of this RFP, addenda will be provided to any Proposer provided the RFP package. CITC reserves the right to postpone the date and time for receipt of proposals at any time prior to the time announced.
- Invoices and Expenses. Independent Contractor will invoice for work on a monthly basis. Independent Contractor will keep an accurate record of all work performed. Invoice must contain an breakdown of work conducted.
- CITC will pay Independent Contractor within 30 days after receipt of approved invoice. CITC will not reimburse for any business-related or travel-related expenses incurred, without prior written agreement.
- CITC's contract term: Ownership of Work Product. All work product/s, equipment or material, including hard or electronic copy, that may be created or purchased under this Contract belong to CITC and Independent Contractor will deliver it to CITC upon termination of this Contract. Independent Contractor agrees that all works of authorship, materials, information, and discoveries created under this Contract is owned by CITC and Independent Contractor assigns to CITC all rights, title, and interests Independent Contractor may have in the works of authorship, materials, information, and discoveries Independent Contractor creates under this Contract, and Independent Contractor retains no rights in any such works of authorship, materials, information, or discoveries. CITC retains the exclusive right to publish, disclose, distribute, copy, license, or otherwise use, in whole or in part, such materials and property. Independent Contractor will execute all papers and perform other proper acts as CITC deems necessary to secure for CITC or its designee the right assigned.
- CITC contract term: Termination Rights. Because circumstances may occur requiring CITC to discontinue a program or project to which Independent Contractors' activities related, it is necessary for CITC to have the right to terminate this contract with Independent Contractor without cause. Therefore, CITC may, at any time, upon fifteen [15] calendar days' written notice terminate Contract without cause.

## RFP Response Requirements

Proposals should include the appropriate narrative and supporting materials to adequately address the scoring criteria. Proposals not containing all of the items listed below may be determined nonresponsive by CITC, and will not be considered.

For consistency and to facilitate evaluation of all responses, proposers must organize their proposals as defined below.

- Part 1
  - Company overview - Background on your business, experience, location, size, capabilities, etc.
  - Why You- What makes your team/solution right for CITC project and needs.
  - Technical expertise - Details on your stack, skills, and methodologies they use.
    - Specifically, call out experience with technology listed in Appendix A: Current Solution Specifications
    - Provide overview of your preferred approach to development that includes CITC staff
- Part 2
  - Proposed solution - Your vision for how you would execute on the project requirements and goals.
    - Include the roles and responsibilities of CITC and vendor.
- Part 3
  - Pricing breakdown - Approach to pricing, rate card by roll, Project management overhead, etc...
    - Rate Card should have detailed information about role, duties, and rate
- Part 4
  - References - List of past client references you can contact for reviews of working with the vendor.
- Part 5
  - Federal contract requirements:
    - Firm's response to Alaska Native/American Indian Preference Requirements.
      - Submit documentation that the Proposer is an AI/AN or Small, Minority, Women-Owned business concern (if applicable).
    - Lobbying Certification and Disclosure.

We look forward to reviewing proposals for how vendors would approach this ongoing engagement to iteratively enhance and expand the capabilities of our participant portal within our Azure environment in a secure and compliant manner.

## Evaluation Process

Proposals received in response to this RFP will be reviewed by the Evaluation Committee. The committee may, at its discretion, decide to interview the proposers.

- An evaluation committee will independently evaluate the merit of proposals received in accordance with the evaluation factors defined in the RFP. Failure of the bidder to provide any information requested in the RFP may result in disqualification of the proposal and shall be the responsibility of the proposer. CITC intends to enter into a contract with the firm which in CITC's sole discretion and judgment will provide the best overall value to CITC. The evaluation process shall be based on a 100-point scale. Categories have been identified for the evaluation process. Each category shall receive a point value within the specified range based on how well the proposal meets or exceeds CITC's requirements.
- Each proposal submitted stands alone and will be evaluated on its own merits in terms of meeting CITC's requirements and terms and conditions, pricing, and overall responsiveness to the Request for Proposal.
- Submission of a proposal implies acceptance of the evaluation technique and recognition that some subjective judgments shall be made by CITC during assignment of points.
- All firms who submitted a proposal will be notified in writing of the results after the scoring and subsequent due diligence is completed.
- Any award as a result of this proposal shall be contingent upon the execution of an appropriate contract. This RFP and its attachments shall form the basis of the Contract Terms and Conditions. **Exceptions or deviations to this proposal must not be added to the proposal pages, but must be on Proposer's letterhead and accompany the proposal.** Any exceptions to the Terms and Conditions will be taken into consideration when evaluating proposals submitted. CITC reserves the right to reject any or all of your proposed modifications.

## **Scoring Criterion**

### **Proposer's approach**

- Narrative approach as described in Purposed Solution of the response;  
***Maximum points 20***

### **Company Qualifications**

- Technical Expertise
- References
- "Why You" Section  
***Maximum points 40***

### **Proposed Pricing**

***Maximum points 30***

### **Proposer Preference**

- American Indian/Alaska Native owned vendor preference. **(10 points)**

## **Award**

Rejection of Proposals: CITC reserves the right to accept or reject any or all proposals, to waive irregularities and technicalities. CITC also reserves the right to reject the proposal of any proposer who has previously failed to perform properly or complete on time contracts of a similar nature, or a proposal from a proposer who, investigation shows, is not in a position to satisfactorily and timely perform the contract.

Selection: CITC desires to enter into negotiations and ultimately reach an agreement with the proposer who demonstrates the best combination of attributes to provide the services required, and who also negotiates fees with CITC that are fair and reasonable. CITC may conduct discussions with any proposer who has submitted a proposal to determine qualifications, for further consideration. Since the initial review by CITC will be deemed preliminary in nature, the document and process will be deemed confidential until such time as the successful proposer(s) is/are selected. CITC is not required to accept the proposal with the lowest costs.

No proposal shall be withdrawn for a period of sixty (60) days subsequent to the deadline date for receipt of the proposals without the written consent of CITC. In no way does this request for proposal constitute a contract, or obligate CITC in any way.

The awarded firm(s) will be required to provide the following:

- Business License
- Certificate of insurance as defined by this RFP
- IRS Form W-9

## **Insurance Coverage**

Proposers must have, at minimum, the following insurance coverage in place, with a company or companies lawfully authorized to do business in the State of Alaska, with coverage written with carriers with an A.M. Best rating of A- VII or better, or acceptable to CITC at its sole discretion, insurance that will protect an Independent Contractor from claims set forth below that may arise out of or result from an Independent Contractor's performance of work and for which Independent Contractor may be legally liable, whether it be by Independent Contractor or by a subcontractor or by anyone directly or indirectly employed by any of them, or by anyone for whose acts any of them may be liable::

1. Commercial General Liability. Coverage with respect to claims, losses, and liabilities that may result directly or indirectly from the work required in the project, with limits of not less than \$1,000,000 each occurrence and \$2,000,000 general aggregate. The coverage will include coverage for premises and operation, products/completed operations,

product liability, blanket contractual liability including coverage for Independent Contractor's contractual indemnities, all of which will be on an occurrence basis. This insurance will provide a waiver of subrogation in favor of CITC and CITC will be named on the policy as an additional insured on a primary and non-contributory basis.

2. Workers' Compensation and Employer's Liability. Workers' compensation coverage as required by law, together with employers' liability coverage, with minimum coverage limits of \$500,000 for bodily injury by disease and/or accident. These insurance policies will waive subrogation in favor of CITC.
3. Commercial Automobile Liability. Coverage with limits of not less than \$300,000 combined single limit, coverage for any owned, non-owned, or hired automobiles, and written on an occurrence basis. This insurance will provide a waiver of subrogation in favor of CITC and CITC will be named on the policy as an additional insured on a primary and non-contributory basis.
4. Network Security / Privacy Liability ("Cyber"). Covering a minimum combined limit of liability of no less than \$1,000,000 each occurrence and \$1,000,000 general aggregate. Coverage will include privacy liability for a breach of any legally protected information whether in electronic or tangible format as well as coverage for a network security event. Coverage will extend to any CITC information that Independent Contractor has agreed in writing to keep confidential. The policy will also cover notification costs as well as credit monitoring for individuals affected by a breach. The policy will name CITC as a co-defendant for any breach of CITC's personally identifiable information in Independent Contractor's care custody and control. This policy will waive subrogation in favor of CITC and name CITC as additional insured on a primary and non-contributory basis.

Additional-insured status will be for all limits carried, not limited to the minimum acceptable as required in a contract.

All coverage other than workers' compensation and employer's liability will apply separately to each insured against whom a claim is made or suit is brought, except with respect to the limit of the insurer's liability.

If any coverage is written on a "claims-made basis," the retroactive/continuity date will precede the start of Work. Coverage will be maintained for a minimum of the duration of applicable statute of limitations/repose or six (6) years, whichever is greater

### **Restrictions on Lobbying Activities**

Entities receiving a federally-funded contract from CITC that is in excess of \$100,000 must comply with the "New Restrictions on Lobbying" found at 24 C.F.R. Part 87. 24 CFR 87.100 provides a baseline prohibition on the use of federal funds for lobbying purposes. 24 CFR 87.110(d) further provides that any person (or entity) receiving a federally funded contract or subcontract, at any tier, that exceeds \$100,000 must file a certification, and, if required, a disclosure form, to the next tier above.

All CITC contractors and subcontractors who will receive a contract or subcontract of \$100,000 or more, to be paid from a federal grant to CITC, must sign and submit to CITC the attached certification form. All contractors and subcontractors who will receive a contract or subcontract of \$100,000 or more must also require all of their subcontractors who will receive a subcontract of \$100,000 or more to sign and submit the required certification and, if applicable, disclosure form.

**Certification for Contracts, Grants, Loans, and Cooperative Agreements**

The undersigned certifies, to the best of his or her knowledge and belief, that:

1. No Federal appropriated funds have been paid or will be paid, by or on behalf of the undersigned, to any person for influencing or attempting to influence an officer or employee of an agency, a Member of Congress, an officer or employee of Congress, or an employee of a Member of Congress in connection with the awarding of any Federal contract, the making of any Federal grant, the making of any Federal loan, the entering into of any cooperative agreement, and the extension, continuation, renewal, amendment, or modification of any Federal contract, grant, loan, or cooperative agreement.
2. If any funds other than Federal appropriated funds have been paid or will be paid to any person for influencing or attempting to influence an officer or employee of any agency, a Member of Congress, an officer or employee of Congress, or an employee of a Member of Congress in connection with this Federal contract, grant, loan, or cooperative agreement, the undersigned shall complete and submit Standard Form-LLL, "Disclosure Form to Report Lobbying," in accordance with its instructions.
3. The undersigned shall require that the language of this certification be included in the award documents for all sub-awards at all tiers (including subcontracts, sub-grants, and contracts under grants, loans, and cooperative agreements) and that all sub-recipients shall certify and disclose accordingly.

This certification is a material representation of fact upon which reliance was placed when this transaction was made or entered into. Submission of this certification is a prerequisite for making or entering into this transaction imposed by section 1352, title 31, U.S. Code. Any person who fails to file the required certification shall be subject to a civil penalty of not less than \$10,000 and not more than \$100,000 for each such failure.

Name: \_\_\_\_\_

Title: \_\_\_\_\_

Signature: \_\_\_\_\_

## **Appendix A: Current Solution Specifications**



# myCITC Technical Specifications Draft

myCITC is an interactive web-based portal designed to facilitate seamless access and interaction between participants and Cook Inlet Tribal Council (CITC) services. The backend was built using C# and ASP.NET Core while the frontend was built using Typescript and React on the Next.JS framework, ensuring reliable performance and a user-friendly interface.

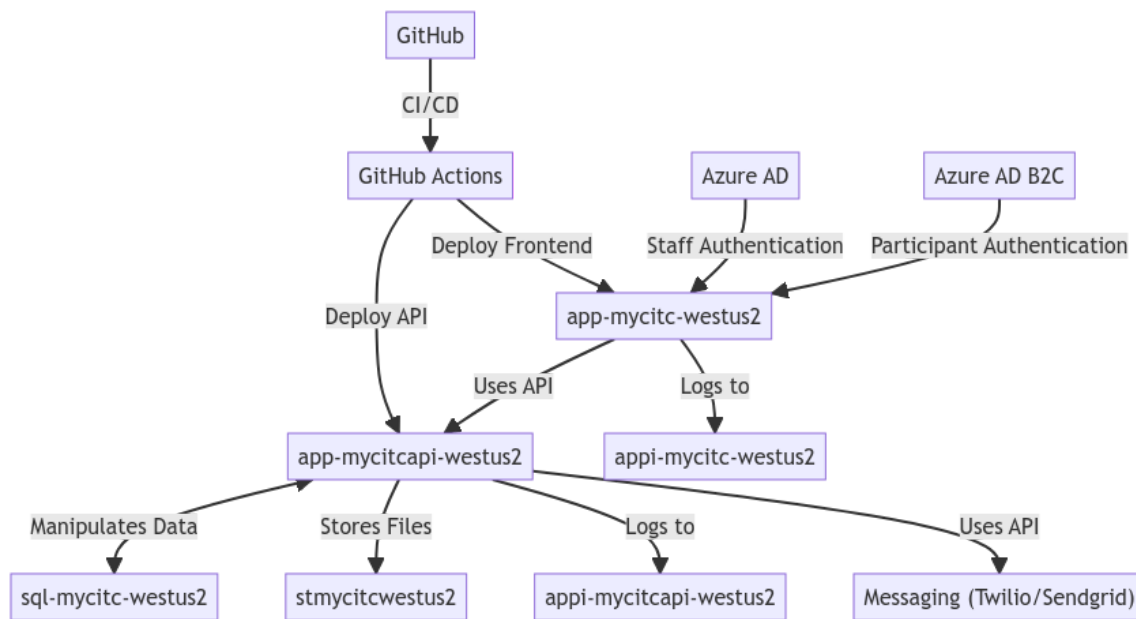
Primary features supported include a comprehensive service directory, participant profile sections, a secure messaging platform, online surveys/applications/forms, and notification system.

Through the myCITC portal, participants can explore CITC's diverse range of services, directly contact relevant service departments, submit demographic information online to facilitate in-person visits, communicate securely with CITC staff, conveniently access necessary forms or surveys, and receive personalized notifications.

The overarching objective of the portal is to simplify and expedite participants' engagement process with CITC's social services, enhancing information dissemination and overall efficiency.

## Solution Details

### High Level Architecture



Basic Architectural Diagram

The high level architecture for myCITC project integrates various robust and reliable services to deliver a seamless user experience:

- **Frontend and Backend:** The frontend and backend of the application are hosted on Azure using Azure App Services.
- **Authentication:** For authentication, we use Azure Active Directory and Azure Active Directory B2C. Azure Active Directory provides a secure method for staff authentication while Azure Active Directory B2C offers a secure and scalable solution for handling participant identities. In addition, we employ JWT (JSON Web Tokens) for secure data transmission between frontend and backend systems.
- **Database:** The SQL databases are hosted on Azure SQL server which reflects scalability, security, and performance efficiency for storing participant data.
- **Storage:** Storage accounts are used to store static files and documents pertaining to the application.

- **Insights:** Application Insights are used to monitor the live web applications, automatically detecting performance anomalies.
- **Communications:** For communications operations, Twilio is used to ensure reliable SMS capabilities and provide a secure messaging platform through their Conversations API. For sending emails, Sendgrid is utilized ensuring high delivery rates and speeds.
- **Deployment:** The deployment of the codebase from GitHub to Azure is managed via GitHub Actions, ensuring rapid, repeatable and reliable deployments.

## Solution Overview

The myCITC portal is a one-stop web platform for Cook Inlet Tribal Council (CITC) participants to easily access services and securely share information. Using state-of-the-art technologies such as ReactJS and ASP.NET Core, the portal combines a user-friendly frontend and a robust backend system.

Participants can navigate services, fill out forms, communicate securely, and receive updates in this portal. The backend handles participant interactions and operations, with data stored securely in Azure SQL databases. Additional features include Azure Active Directory for authentication, Azure Blob Storage for file handling, Twilio and SendGrid for communication, and Azure Application Insights for proactive performance monitoring.

## Technology Stack

The myCITC portal utilizes a variety of advanced technologies and libraries to deliver a user-friendly and secure online experience. Here's a detailed look at these technologies:

### Frontend Technologies:

- **ReactJS:** A powerful JavaScript library utilized for building high-performance user interfaces.
- **Next.JS:** A React framework that enables features such as server-side rendering and generating static webpages for React-based web applications.
- **TypeScript:** A statically-typed superset of JavaScript that enhances code reliability and maintainability.
- **DaisyUI:** A plugin for Tailwind CSS that is used for styling the app.
- **Axios:** Promise-based HTTP client for the browser and Node.js, utilized for making asynchronous HTTP requests to REST endpoints.
- **SWR:** A React Hooks library for remote data fetching.

### Backend Technologies:

- **ASP.NET Core:** A cross-platform, open-source framework used for building modern, internet-connected applications with improved performance metrics and simplified coding structure.
- **C#:** A multi-paradigm programming language encompassing strong typing, imperative, declarative, functional, procedural, and object-oriented disciplines.
- **Entity Framework Core:** An open-source, lightweight, cross-platform version of the popular Entity Framework data access technology from Microsoft.

### Database Technologies:

- **SQL Server:** A fully-managed and scalable relational database management system that supports structured and unstructured data.

### Testing Frameworks:

- **xUnit:** A free, open-source unit testing tool for .Net that is used in the backend for automated testing of the code.
- **Cypress:** End-to-end testing framework used in the frontend to write and run tests in real browsers.

### Authentication:

- **Azure Active Directory:** Used for staff authentication, offering secure and compliant methods for authenticating users.
- **Azure Active Directory B2C:** A customer identity access management solution used for participant's authentication.

### **Storage:**

- Azure Blob Storage: Used for storing static files and documents related to the services offered by CITC.

### **Performance Monitoring:**

- Azure Application Insights: A service used to monitor live web applications, automatically detecting performance anomalies.

### **Communication Services & APIs:**

- Twilio: A cloud communications platform that provides messaging, voice, and video communication allowing integration into software applications via APIs.
- Sendgrid: A cloud-based email delivery service used for sending emails.

### **Deployment:**

- Azure App Services: Used to host the frontend and backend of the myCITC portal.
- GitHub Actions: Manages deployment of the codebase from GitHub to Azure to ensure rapid, repeatable deployments.

### **Development Stack:**

**Version Control:** Git, via GitHub, is leveraged for version control, ensuring effective tracking and management of changes to the codebase.

**Integrated Development Environments (IDEs):** Our development team is flexible in terms of IDE choice. We use JetBrains products such as Rider, DataGrip, and WebStorm, along with Visual Studio products including Visual Studio and Visual Studio Code.

**Pre-Production Environments:** We employ Azure App Service slots for our testing and staging environments, allowing us to safely validate changes before pushing to production.

**Build Tools:** .Net build tools are utilized for building our C# backend. For our frontend, we use yarn, a powerful package manager supplementing our use of Next.JS.

**Testing Tools:** Alongside Cypress and xUnit mentioned earlier for end-to-end and unit testing respectively, no other specific testing tools are currently being used.

**Dependency Management:** Our dependency management relies on yarn for handling our JavaScript libraries in the frontend, while NuGet is used for handling .NET packages in the backend.

**Code Review Process:** The primary tool for our code reviews is GitHub Pull Requests which assists us in conducting thorough code reviews and maintain high-quality code hygiene.

**Styling & Linting:** Prettier is employed as a code formatter in the frontend. For C#, while there's no strict linting tool, we adhere to best practices and carry out regular code reviews to ensure consistency.

**Containerization:** Docker is employed in the build test stage of the backend, ensuring application runs seamlessly in an isolated environment.

**Continuous Integration / Deployment (CI/CD):** We utilize GitHub Actions for managing deployment operations from version control to Azure. Deployment practices are streamlined with a blue-green deployment strategy using Azure Slot Swapping. This setup enables us to maintain regular, reliable deployment processes and minimizes the risks associated with the deployment of faulty or unstable code.

**Code Quality/Security Scanning Tools:** For ensuring code quality and security, we use Dependabot, a native GitHub tool for dependency updates and security fixes. This tool helps us identify and mitigate potential security threats in our dependencies promptly.

## **Development / Deployment Pipeline**

### **Environment Setup and Management:**

The different environments for myCITC portal, namely test, staging, and production, are set up in Azure App Service Slots. MyCITC leverages the unique flexibility of Azure's deployment slots to create a workflow that suits its development cycle: work is first previewed and tested in the test environment, before being moved to a staging environment for final checks, and eventually being deployed to production.

### **Pipeline Strategy:**

The myCITC pipeline strategy is completely driven by GitHub Actions workflows. The workflows properly segregate tasks for the front-end and the back-end. The entire process is streamlined, with tasks for build, test, and deploy for both front-end and back-end jobs. The multi-stage workflows enable easy tracing of any issues as well.

### **Artifact Management:**

Artifact tracking and storage are managed by GitHub Actions, facilitating the easy transition between different stages of the pipeline.

### **Trigger Methods:**

Deployment to the test environment occurs automatically upon a merge into the main branch. However, deployments to other stages like staging and production require approval from a tech lead. This method ensures robust checks at critical stages while maintaining the speed of deployment processes.

### **Rollback Procedures:**

If any anomaly or problem is detected during the course of the development pipeline, we are equipped for quick rollback procedures. In case of deployment issues, an earlier version can be redeployed. If any database changes introduce errors, we can rollback the latest migration using Entity Framework Core.

### **Change Management:**

All changes to the process or tools in the development pipeline are managed through GitHub Actions workflows. Any potential changes are evaluated by our change advisory board (CAB) before deployment to production.

### **Production Release Management:**

The production release strategy involves moving from a test environment to staging and finally to production. The transition from staging to production is coordinated via manual approval from a tech lead and is facilitated by Azure's slot swapping capabilities. This ensures that there's a solid safety net to catch any issues before they reach the end-user.

## **Plugin/Module Capabilities**

The myCITC platform is designed to be highly modular and flexible, with robust plugin capabilities for both the frontend and backend.

### **Core Capabilities**

Both frontend and backend have the capacity to dynamically load plugins via packages, either from package repositories, local sources, or remote locations. These plugins can be seamlessly integrated into the existing functionalities of the myCITC portal and can respond to events triggered within the core website.

### **Frontend Plugin Capabilities**

The frontend plugin capabilities enhance the overall user interface and navigation of the myCITC portal:

- **Adding New Page Links:** Frontend plugins have the provision to introduce new page links in navigation areas.
- **Creating New Pages:** The plugins can add new pages to the UI, complete with new business logic, validation, etc.

### **Backend Plugin Capabilities**

The backend plugin capabilities cater to a more diverse range of modifications that contribute to expanding and upgrading the functionality of the portal:

- **Adding New Endpoints:** Backend plugins have the capacity to add new endpoints, thereby extending the realm of operations possible through the myCITC platform.

- **Implementing New Database Tables:** Plugins can also introduce new tables into the database where each plugin-specific schema helps maintain organized, segregated data storage.
- **Reacting to API Events Notifications:** The backend plugins can register as observers for API event notifications, thereby responding promptly as per their intended functionality.
- **Replacing Existing Functionality:** An exceptional feature of the backend system is its ability to allow plugins to replace an existing feature with its new implementation, contributing to continuous optimization of functionalities.

The plugin/module capabilities currently being utilized are relatively straightforward, yet they provide immense extensibility possibilities for the myCITC web application. As capabilities evolve over time, these expansions will continue to accommodate newer developments, thereby maintaining the platform's contemporary appeal, functionality, and reliability.

## Development Practices

Our development practices rely heavily on a combination of Kanban and Scrum methodologies, ensuring efficient task management and rapid response to changes in demand or priority. Here's an overview of our development practices:

**Methodology:** We've found that a combination of the visual, workflow-centered approach of Kanban and the iterative, incremental work cycles of Scrum best facilitate our development process.

**Solution Updates:** To ensure we consistently deliver top-quality performance and functionalities, our team performs regular updates to our application, deploying to production anywhere from one to four times a week.

**Requests and Bug Management:** Once confirmed and vetted, feature requests and bug reports are added to our backlog. From there, they're seamlessly integrated into ongoing development work.

**Deployment Schedule:** Our deployment schedule showcases flexibility to meet varying project needs. The test environment is continuously deployed, while more substantial updates are reviewed during weekly Change Advisory Board (CAB) meetings. Beyond this, we maintain a fluid release schedule to accommodate urgent fixes or features.

**Code Quality Assurance:** Despite not having an explicit set of coding standards, our team ensures high code quality by adhering to accepted best practices in development. Moreover, we execute thorough code reviews and facilitate collaborative problem-solving sessions as needed.

**Project Management Tools:** We employ the Atlassian suite for comprehensive project management functionality. This includes tools such as Jira for issue and project tracking and Confluence for documentation.

In essence, our practices place a high emphasis on frequent iterations, teamwork, constant communication, and a steady stream of improvements to ensure top-quality software solutions for our clients.

## Testing Practices

**Types of Testing Performed:** We primarily focus on automated Unit and Integration testing. However, our developers also perform hands-on testing of new features in the test environment to validate functionality and usability.

**Testing Tools:** On the backend, we utilize xUnit for testing our APIs; while on the frontend, Cypress is our tool of choice.

**Automated Testing:** Every pull request triggers an automated test suite to ensure code integrity before merging. These tests are also run when PRs are merged into the main branch.

**Bug Tracking:** Bugs detected during feature testing or post-deployment are logged in Jira for tracking and resolution. All bugs are examined, confirmed, and assigned to the appropriate team member for timely resolution. We prioritize these bugs based on business impact, visibility, and the level of effect they have on end-users.

**Quality Assurance:** Our commitment to quality assurance is unwavering. It starts with stringent automated testing, extends into manual feature validation, bug tracking and resolution, and encompasses a continuous improvement mindset where every bug identified is another opportunity for improvement.

**Performance Testing:** At present, there's no formal policy on performance or stress testing practices aside from monitoring of high-traffic pathways. However, we are actively exploring viable methods to integrate these testing forms into our pipeline.

**Regression Testing:** Our automated testing setup aims to guard against regression issues by being tied into our version control and deployment processes. It ensures that changes introduced with new feature deployments do not break existing functionalities.

**Test Coverage:** We don't strictly enforce code coverage metrics right now but we can generate reports to review areas that lack test coverage. This allows us to continually reevaluate and improve our testing procedures over time.

## Monitoring and Alerting Practices

We utilize Azure Application Insights to monitor our application in real time. This includes tracking custom events, tracing logs and exceptions, and analytics. This facilitates the tracking of system performances and critical operations of the myCITC portal.

Azure Application Insights also provides us with the ability to visualize and analyze the application's telemetry data. This helps us to diagnose issues and understand what users do with our app.

Even though the primary monitoring tool is Azure Application Insights, we are currently exploring best practices for setting up alerting systems that will notify the team in case of system failures or critical exceptions. This will allow us to respond to and resolve issues faster and ensure the stability and reliability of the myCITC portal for our users.

Furthermore, GitHub notifies us of any security threats related to our dependencies. We regularly monitor these alerts and take necessary action to mitigate any risks promptly.

In general, all logs are retained and available for inspection. The system has extensive logging of application flows and exceptions, assisting in prompt identification and resolution of issues.

## Known Risks or Issues

### Architectural Debt

During the initial development phase, expedited delivery resulted in the accumulation of architectural debt, particularly in the backend. Addressing these issues is now a priority.

### Endpoint Routing

The current state of endpoint routing diverges from RESTful API best practices in some instances. The plan is to refactor these routes to align with accepted standards.

### Documentation

There is a marked need for enhancing the quality and comprehensiveness of our technical documentation to facilitate better understanding of the system and its components.

### Mitigation Plans

- For architectural debt, the plan is to allot dedicated maintenance periods for refactoring and improvements.
- In terms of endpoint routing, upcoming revisions will focus on aligning all routes with RESTful practices.
- For documentation, systematic efforts are underway to incrementally improve our technical write-ups, ensuring clarity and comprehensive coverage of the system.

No known security or scalability issues are identified at this moment.