I. Portions of the functionality that have already been written. Use of the functionality described below is optional to the vendor, and offering access to them is intended to speed up development time, as well as providing a smoother transition for long-term support of the application to IDALS IT.

A. eMail integration

B. User authentication

C. Digital signatures

D. Map integration

II. PHP email function available for use

A. by including a PHP file (supplied to the vendor selected for the RPF), you would have access to the following function:

1. `function idals_email($to, $toname, $subj, $msg, $from='agri');`

   a) `$toname` = *user's full name (name appended to the email in the* `to`: *line)*

   b) `$to` = *user's email*

   c) `$subject` = *subject of the email*

   d) `$msg` = *HTML formatted email*

   e) `$from` = *can be omitted unless you want to use a different account to send from.*

      (1) Default is agri@iowaagriculture.gov

      (2) Adding chooseiowa@iowaagriculture.gov is an option, but we would need to include IDALS' email administrator to help implement this

III. User Authentication

A. Requirements for use

1. There are 6 PHP pages in the `auth` directory that can be used dynamically for any site hosted on the server. IDALS IT will need to add a record to the `app` table of a shared database for the dev, test, and production environment for the application.

2. These files **may not be altered** by the vendor. IDALS IT will set permissions such that this cannot happen

3. This requires the MySQL database to have a **`user`** table with certain fields. IDALS IT will implement this in collaboration with the selected vendor.

4. A PHP header include file must be created by the vendor with some boilerplate code supplied by IDALS IT. A site-wide style sheet should be included in this header file with a `link` tag. There is also a footer file, but that file is very basic

B. Files

1. index.php

   a) *This is the main login page that has a username & password field along with buttons/links for*

      (1) Forgot Password

      (2) Sign-up for a new account.

(3) Okta integration button (see below)

2. forgot.php
   a) *This allows the user to send themselves a "forgot password" email with a link to reset.*
   b) *The `user` table is updated with a key, source IP, and reset timestamp*
3. reset.php
   a) *This is the target of the "forgot password" link that gets email.*
   b) *it checks the key & the source IP as well as checking the reset timestamp for age.*
   c) *If the above checks pass, the user is given the opportunity to set a new password and login again (back to the index page)*
4. new_acct.php
   a) *A user can create a new account if the application is set up for self-created accounts.*
5. profile.php
   a) *A user updates their profile - **THIS IS NOT GOING TO BE USED FOR THIS APPLICATOIN**. Choose Iowa will have a participant profile that is to be developed by the vendor as a part of the scope of this project*
6. okta.php
   a) *An Okta integration was created and tested against the state's Okta text/dev environment, but not implemented in projection yet.*

IV. Digital Signatures

A. IDLAS IT developed a digital signature for some internal projects.

B. The digital signature is written in PHP with a MySQL back-end database.

1. On the front-end, there are 4 parts to the signature at the bottom of the form.

Area to type signer's name	Checkbox for disclaimer
A stylized version of the name (visual signature)	Button "Sign and Submit"

C. On the back-end, the following data elements are stored as a part of the signature

1. Signature Name (from the top-left quadrant of the front-end above)
2. Boolean for the disclaimer checkbox
3. IP address of the source of the client computer
4. Timestamp of the signature
5. The hash of the form data
6. If the user is authenticated, the user_id of the authenticated user.

D. How the digital signature works:

1. When the form is signed and submitted, all the data contained within the form is concatenated. A timestamp is also recorded and added to the pre-hash data.
2. After concatenation, a hash is applied to the data, and that hash is stored along with the form data (see section C above)
3. Whenever the form is reviewed, the same hash is applied to the data currently stored, and if the hashes match, we know that the data has not been altered since the document was signed.

E. The code for the digital signature can be repurposed for this implementation provided the code structure is similar enough to the way internal applications have been developed.

V. An integration with Google Maps has been created for this project.

A. A small amount of work/effort has been put into this. It may or may not be useful to the vendor

B. The code-base for this integration will be made available to the selected vendor