Go Code Colorado 2018 Technical Evaluation Criteria

User Experience (30%)	
The product has no working defects and is user friendly, and the experience is interactive and displays dynamic data.	4
The product is built to a working product standard, can be navigated, but has minor bugs. Data experience is interactive and displays dynamic data.	3
The product is built to a working product standard, but is not easily navigated and has many bugs and/or is missing key features. Data experience is interactive with static data.	2
The product is difficult to navigate, and the design makes it difficult to add features. Data experience is static.	1
The product is not functioning or available on any test or live server, app store or embedded on a website. No data displayed.	0

Sustainability (30% of total score)	
The product is extremely well documented, comments are consistent, and no grammar or spelling errors. Product is built on software or platform with active support and requires no manual updates. Examples: Angular 1 vs 2 or NodeJS 0.10 vs LTS.	4
The product is reasonably well documented, minor formatting omissions, and limited grammar or spelling errors. Product is built on software/platform/libraries with active support and requires minimal manual updates.	3
The product is marginally documented, significant portions are documented incorrectly, and has a significant number of spelling and/or grammar errors. Product is built on software or platform that is no longer supported/deprecated.	2
The product is poorly documented. There are minimal comments and/or the comments are incorrect. Product is built on software or platform that has been forked/modified from supported/original source.	1
No documentation, product corrupted without frequent maintenance, and/or code is not properly licensed.	0

Functionality (20% of total score)	
The product is completely functional and responds correctly under all functional tests producing the correct responses and the data is represented correctly.	4
The product is mostly functional and responds correctly under all functional tests producing the correct responses and the data is represented correctly with acceptable obfuscation.	3
The product is marginally functional with numerous errors. The product may respond correctly under certain circumstances, but there are significant errors, incomplete code sections, or the data representation is obfuscated.	2
The product is minimally functional with significant portions of the code missing or incomplete. The product is largely non-responsive to most functional tests, and the data representation is clearly incorrect or otherwise distorted.	1
The product is not functional, meeting no significant design specifications, and/or the interface does not display data.	0

Logical Structure (10% of total score)	
Documentation and code are extremely well organized, properly formatted, and related code sections are logically grouped. Data is optimally stored and organized as appropriate and data architecture/schema are well documented.	4
Documentation and code are easy to follow with logical groupings of related code, but minor formatting problems. Data stored in web server/file system, data architecture documented. Or inversely, data is optimally stored but no documentation.	3
Documentation and code are readable only with significant effort, and there is little to no formatting and/or significant problems with its organization. Data is only available on a local machine, and updates are manual. Data architecture is poorly documented.	2
Documentation and code are poorly organized and difficult to read without consistency in formatting and logical code grouping. Data setup logic is unclear, and no data architecture documentation.	1
Documentation and code are readable only by someone extremely knowledgeable with its layout and purpose. No data use.	0

Security (10% of total score)	
Team has taken measures to secure the product by finding, fixing and preventing security vulnerabilities. These include Input Validation, Authentication, Authorization, Sensitive Information, Session management, and Parameter manipulation.	4
Product has shown a deficiency in fully preventing known security vulnerabilities.	3
Exposed to some common vulnerabilities that risk breach of sensitive, confidential, proprietary or personally identifiable information (SQL injection, calls to OS through command shells), or tool embedded in website that is insecure.	2
Authentication mechanism without adequate protections (e.g., clear text passwords), website or RESTful API not secure.	1
Required mechanisms absent. Lack of "Terms of Use", "Privacy Policy", information sent in clear text and not over SSL.	0