App Name

Categories	Developing	Satisfactory	Proficient	Exceptional	Comments
Professional Skills	1	2	3	4	
Ambition/Hustle				Pop pitch articulates ambitious and achievable plans for short- and long-term achievement	
Self-Awareness/Willingness to Ask for Help				Reaches out for help proactively and tactfully, can identify supporters for different needs.	
Work ethic				Consistently puts in the work to accomplish project milestones, including demo.	
Attention to detail			Presentation follows the Access Code template and contains no visual distractions (typos, visual formatting)	Recognizes details as important to contributing to overall quality of work product and executes accordingly.	
Drive for excellence				Sets very high standards for self in work - usually meets these standards and thoughtfully responds when standards are not met	
Leadership				Teammates look to this person as a role model. Provides direction in making groups of all sizes work together better.	
Teamwork: Demo				Project demo runs smoothly with speaker and driver roles well-orchestrated	
Teamwork: Dynamics				Team roles are well-designed and conflicts are addressed in a timely and professional manner via norms as agreed upon by the group	
Documentation [Portfolio]				Published project is updated on resume, LinkedIn, forked to individual GitHub, and addressed in cover letter as relevant	
Project Management				Trello Board is accurate at each weekly check-in meeting and team can reallocate tasks responsively.	
Professional Workplace Skills			Team consistently presents agenda at weekly Tech Mentor check-ins and sends notes as recap		
Professional Workplace Skills: Public Speaking		Demo requires memory aids; content is delivered clearly but verbal and non-verbal presentation skills are not apparent		Confident and well-rehearsed delivery without memory aids; verbal and non-verbal presentation skills enhance the delivery of the content	
Tech Fluency	1	2	3	4	
Demonstrated Passion for Technology: User Story	Product demo does not address potential end users			Product demo clearly illustrates how product fits into compelling user story/stories	
Demonstrated Passion for Technology: Product/Market Fit	Project presentation does mention existing products	Project presentation mentions existing products nominally	Project presentation alludes to notable features of existing products	Product demo illustrates understanding of existing products and how product fits in or improves upon offerings	
Demonstrated Passion for Technology: Go to Market Strategy	Project presentation does not address user discovery	Project presentation mentions generic ways for the product to be discovered by potential users	Project presentation mentions a few insightful ways for the product to be discovered by potential users	Project presentation presents a comprehensive "go to market" strategy for the product	
Demonstrated Passion for Technology: Innovation			Project demonstrates independent research into areas not explicitly taught in Units 1-6 of Access Code	Project demonstrates successful implementation of independent research into areas not explicitly taught in Units 1-6 of Access Code	
Development of Social Capital			Fellows are connected to judges (Hackathon), panelists (Demo Day) on LinkedIn or other social media as relevant (e.g. Twitter)	Fellow has connected 1:1 in-person with a member of the tech community as an outcome of this project, e.g. for an information interview or consultation on project direction	
Technical Mastery	1	2	3	4	
Portfolio Suitability					

App is published and accessible on the web/App Store/Play Store	Not published	Partially published (e.g. doesn't work all the time, some parts inacessable)	Fully published (OK if in "beta" mode)	Fully published w/ great "marketing" (welcome page/message, description. screenshots/GIFs, links to devs, etc.)	N/A for hackathon OK to use TestFlight for App Store, alpha/beta tests for Play Store
App builds/compiles successfully	No	N/A	N/A	Yes	
App runs without crashing/hanging	Crashes on startup	Crashes on some use cases (not just edge cases)	Minimal crashes, only on edge cases (not on any main use cases)	No crashes	
Uncontrollable errors (e.g. server errors) are handled gracefully, logged, explained to user	App crashes on these errors	Some errors handled, no logging or user notification	All errors handled, most logged & user notified descriptively	All errors handled, logged, explained descriptively	
-					
GitHub Usage					
Source code hosted on GitHub with a history of commits with clear messages	Only one or a few commits, poor messages	Many commits, but poor messages, some commits of broken code	Many commits w/ good messages, and no commits of broken code	Many commits w/ good messages, no broken code, clear use of branching	
Commits are made frequently and are each focused on a single feature	Commits are infrequent and all touch multiple app features	Some commits are focused to a single feature	Majority of commits focus on a single feature	All commits focus on a single feature	Possibly N/A for hackathon
README.md file is well-formatted (markdown) and explains project thoroughly	No README	README exists w/ minimal info, poor formatting	README has good info, readable formatting	README has great info, beautiful formatting, including screenshots/GIFs	Probably low expectations for README during hackathon, but should be added later
Code Quality					
Code follows D.R.Y. principle	>=5 instances of code repetition	<5 instances of code repetition	<3 instances of code repetition	0 instances of code repetition	
Variable/Class/package names are descriptive and follow naming conventions	<50% consistent & descriptive	50% consistent & descriptive	80% consistent & descriptive	100% consistent & descriptive	
Code shows clear division of responsibilities by class/package/file	Little or no division of responsibilities	Majority of code units (classes/files) have single responsibility	All code units have single responsibility, but not totally adhering to a specific architecture pattern (aside from default MVC)	All code units have single responsibility and app adheres to an architecture pattern (MVP/MVVM/Clean/etc)	
Code adheres to relevant style guide and is clearly commented/documented	True for <50% of code	True for 50% of code	True for 80% of code	True for 100% of code	
Design and Planning					
App is focused on solving a specfic problem, not a collection of unrelated features	No working features	Working features, but not all consistently addressing the same problem	All features work and clearly address some aspect of the problem to be solved	App provides a creative and thorough solution to the given problem, with all aspects of the problem addressed and no unrelated features	
App implements clearly documented user stories	No user stories	Some user stories, but no clear connection to app features	Most app features directly relate to a documented user story	All app features directly relate to a documented user story	
Intuitive UX design reflecting evidence (notes) from user testing	Totally unclear UX	Decent UX, but no evidence of user testing	Clear UX with evidence of user testing	Clear UX with obvious ties to evidence from user testing	Possibly N/A for hackathon
Modern, sophisticated UI design (e.g. Material Design for Android)	Broken UI	Mostly functional UI	Fully functional, if not modern, UI	Fully functional and modern UI	Possibly N/A for hackathon
App is responsive to device screen size/orientation	Does not work form some sizes/orientations	Works, but some content unreadable or unclickable	Works w/ all content readable and/or clickable	Separate UI flow for different sizes/orientations to provide consistent UX	Each stack to provide footnote with detailed expectations (e.g. which browsers supported for web, which specific devices/OS versions for iOS/Android)
Functionality					
App retrieves data from a remote API via HTTP calls	No API calls	API calls intertwined with UI code and/or not all asynchronous	API calls separated from UI code and asynchronous	API calls separated from UI code and asynchronous, and using authentication (API key or OAuth)	
App does something meaningful with data from API (transform/format/analyze/etc)	No API data used	API data is displayed with no work performed	App performs work on API data to improve utility to user	App combines multiple APIs and/or gathers user data to create a whole new dataset	
App persists data in a database	No database	DB code intertwined with UI code and/or not all asynchronous	DB code separated from UI code and asynhronous	DB code separated from UI code and asynhronous w/ user authetication	
Automated unit testing and/or UI testing	No testing	<50% of features tested	>50% of features tested	All features tested	N/A for hackathon