Project Evaluation Document

Aaron, Gabe G., Gabe L., Nick

In this document, we seek to reflect upon our experiences this semester working together as a team to create Solien Gallery. While there were certainly moments where more foresight could have been used and where clearer communication could have been helpful, we can say confidently that we are proud of what we have been able to accomplish in this semester. Learning a diverse set of topics as were required for this project, ranging from image creation to blockchain technologies to background tasks, has been an experience that was both immensely challenging and rewarding. With that in mind, we will reflect upon (1) *our original planning and our milestones*, (2) *our experience with design and systems used, interfaces, languages, and testing*, (3) *some surprises we experienced*, (4) *some choices we made that worked well and badly*, and (5) *what we would do if we had more time and what we would do differently next time*.

Regarding our planning and milestones, we very successfully differentiated our MVP from our stretch goals. We defined our MVP concisely very early on, and successfully implemented it before the beta was due. As a result of that, we were able to implement an optimizing stretch goal, namely an entirely newly constructed local database (with the potential to grow dynamically in the future) with Solien NFT metadata. Looking at that "Project Overview" document that we were required to compose earlier in the semester, we accomplished all 9 milestones of the MVP, more or less in the proposed order, in addition to the first stretch goal. This was something we did not necessarily expect to accomplish, especially given that our discussions at the beginning of the semester with the course staff regarding our topic generally produced the consensus that our project was exceptionally risky and in need of serious research relative to standard COS333 project ideas. This was only possible because of the investment in time that we made at the beginning of the semester to thoroughly think through our project's steps sequentially, as well as focusing on conceiving of its architecture.

It is perhaps with regard to design and systems used, interfaces, languages, and testing that we learned the most this semester. For most of us, this was the first full stack experience that we have had in creating an app. As such, we were challenged to think in new and creative ways. The fact that we spent so much time at the beginning of the semester drawing up the architecture of our systems and the full stack designs we made prior to our project approval meeting reflects this. Interfacing early on proved to be immensely helpful to us in the long run. Specifically, our implementation of img_util.py and

solana_util.py allowed for succinct communication between different parts of our application where a “request” and a “response” were needed for both blockchain related data and image processing related data. Doing this at the beginning of the semester made things progress much more quickly later on in the semester. Additionally, many of us learned new languages this semester, either from scratch or from a relatively rudimentary point. This list includes CSS, JavaScript, HTML, and programming with JSON. The portion of the team mostly focusing on the backend even learned some Rust when watching tutorials relating to blockchain traversal. For all of us, this was a rewarding experience to be able to practically apply this learning. And in terms of testing, we had the opportunity to develop our own tests based on the specific needs of our system and in accordance with many of the techniques described in lecture. This was an essential part of the end-to-end development experience.

Throughout the semester, there were definitely some serious surprises encountered. For example, although we expected things to be complicated, figuring out how to search the blockchain proved to be much more difficult than expected. As a result of this, while we certainly learned a lot, there was a period of time relatively early on in the semester where we made little progress towards the goal of retrieving data we could understand from the blockchain. Another surprise we encountered emerged from working with Heroku. We unfortunately discovered that Heroku times out requests after 30 seconds, and this amount of time was necessary for many blockchain searches simply because of their computational complexity (especially prior to the development of our caching system). As a result of this surprise, we added loading pages in our architecture to delegate the time-consuming operations to background tasks.

In addition to thinking about the surprises we encountered, it is also important to reflect on some of the things that went well and some of the things that didn’t. Some of the principal successes in our project include our decision to use interfaces early on, our caching system which greatly optimizes our performance, our background task system via our loading pages, as well as developing many of the aesthetic features of our website, including the features allowing for gallery customization. These features all have in common the trait that they improved the operation of our website while at the same time making it more user friendly. On the other hand, certain things did not go as planned. For example, and as previously alluded to, we experienced a lot of difficulty in getting data from the blockchain at first. The portion of the team focusing on the backend spent a lot of time working to get data from the blockchain based on hundreds of lines of JavaScript code from very new libraries we discovered. This involved doing things that required a serious deal of learning, including adapting these hundreds of lines of code to our purposes in addition to working with TypeScript conversion. We actually may have been extremely close

to achieving our goal with this more difficult method, but we were stuck at a point where we needed to decode data being returned to us in an usual encoding. Ultimately, we discovered a more elegant API to work with for this purpose which we have chosen to use. We learned from this that it is essential to build upon existing work when possible rather than trying to work on things only from scratch. Additionally, one other way in which things didn't go as planned was with regard to the web frameworks we chose to use. At first, we implemented a skeleton of our project in Express, but we ultimately used Flask once we changed our backend language from JavaScript to Python. It proved to be worthwhile to be willing to change such a fundamental feature of our application once we learned that it had many potential benefits.

More generally, another thing that we think went well was the success in fulfilling our stated purpose. In the "Project Definition" section of the "Project" page of the course website, it is written that "Some of the best projects come from noticing a task that is done by hand or poorly by machine, when it could be done really well by a suitable program, or where something complicated could profit from a neat user interface. Think about how rapid interaction might be used in some new application, or how some standalone program could be web-ified, or how some service could work on a phone. Or maybe focus on tools that make it easier to build such things, and create a couple of examples that demonstrate the wonders of your tools." We noticed that NFT owners were performing a task manually that was in need of web-ification, namely creating Twitter banners merely out of their NFTs. NFT owners tend (for now!) to simply use an aesthetically displeasing screenshot of a single NFT that they own, so creating an elegant platform to create Twitter banners out of NFTs emerged as a necessity. In response to that, we attempted to do exactly what that excerpt suggests makes a good project by taking the opportunity to improve an existing process, and we successfully did so.

Provided with the ability to do things over again or with more time on this project, there are a few changes we would look to make. For example, if we could do things over again it would have been great to use from the beginning some of the libraries we found throughout the semester that were successful and to have always used Flask instead of Express. Additionally, it would have been helpful to begin thinking about deployment requirements from the beginning of the semester, which would have allowed us to foresee the need to delegate tasks to background tasks on Heroku. If we had more time on this project, we would have enthusiastically attempted to make progress on our other stretch goals, such as adding an authentication system to confirm that the input wallet is that of the particular user. While this was an especially challenging task given the open nature of the blockchain, it would have nonetheless been something great to work on if we had more time. In fact, we plan on working on this after the course ends!

Overall, we are very proud of what we learned and what we accomplished in this project. Whether it was learning how to design an end-to-end system and work together as a team on a project like this or learning more technical components of blockchain traversal and background tasks, working on Solien Gallery provided us with the tremendous opportunity to learn and to reflect on our progress in this project. We look forward to continuing working on Solien Gallery after this semester ends, and we also look forward to applying what we have learned through these experiences to future endeavors. Thank you for a great course!