Reviewers: Jaiveer, Derik, Jasleen, Viet

Code Review Checklist

General

- Does the code work? Does it perform its intended function, the logic is correct etc. **YES**
- Is all the code easily understood? **YES**
- Does it conform to your agreed coding conventions? These will usually cover location of braces, variable and function names, line length, indentations, formatting, and comments. **YES**
- Is there any redundant or duplicate code? **NO**
- Is the code as modular as possible? **YES**
- Can any global variables be replaced? **YES**
- Is there any commented out code? **YES**
- Do loops have a set length and correct termination conditions? **YES**
- Do the names used in the program convey intent? **YES**

Performance

- Are there any obvious optimizations that will improve performance? **NO**
- Can any of the code be replaced with library or built-in functions? **NO**
- Can any logging or debugging code be removed? **NO**

Security

- Are all data inputs checked (for the correct type, length, format, and range) and encoded? **YES**
- Are output values checked and encoded? **YES**
- Are invalid parameter values handled? **YES**

Documentation

- Do comments exist and describe the intent of the code? **YES**
- Are all functions commented? **YES**
- Is any unusual behavior or edge-case handling described? **NO**
- Is there any incomplete code? If so, should it be removed or flagged with a suitable marker like ‘TODO’? **NO**

Testing

- Is the code testable? The code should be structured so that it doesn’t add too many or hide dependencies, is unable to initialize objects, test frameworks can use methods etc. **YES**

- Do tests exist, and are they comprehensive? **YES**
- Do unit tests actually test that the code is performing the intended functionality? **YES**
- Could any test code be replaced with the use of an existing API? **NO**