PBI - description	Time Estimation	Tasks	Not Started	In Progress	Completed
1. Ballot Reassignment for STV : As an election official, I want ballots should be assigned in the correct order during STV elections, so that candidates are elected or eliminated based on fair ballot redistribution.	2 hrs	Debug STV - Implement STV class functions redistributeSurplus(), findLowestCandidate() ; Refactor redistributeEliminated( ), runElection() of STV class; Perform System Tests, Unit Tests, Documentation			Debugged STV - Implemented these functions in STV class - redistributeSurplus(), findLowestCandidate(), and refactored the functions - redistributeEliminated() and runElection() of STV class, Anwesha, System Tests with valid and invalid ballots- Anwesha Added Unit Tests - Zoe/Anwesha, Documentation - Anwesha
2. One Candidate in STV: As an election official, I want the STV election algorithm to function when there is only one candidate, so that even rare election scenarios produce fair and accurate outcomes.	2 hrs	Debug STV → Refactored runElection() and findLowestCandidate() in STV class; Complete System Tests with 1 candidate in STV CSV; Unit Tests in stv_UT.cpp; Documentation			Completed debugging STV-> refactored the runElection() and findLowestCandidate() functions of STV class - Anwesha. Completed System Tests with 1 candidate in STV csv - Anwesha, Unit Tests - Anwesha/Zoe, Added inline Documentation - Anwesha, Doxygen - Zoe
3. STV Results Output: As an election official, I want all candidate names to appear in STV election results, so that no candidate is omitted from the winners/losers list.	2 hrs	Debug Election class output functions – printToAudit(), generateResultsText() , displayResults() to fix STV result output issues; log invalid ballots in display and audit file by updating stvballot error message; complete System Tests ; Unit Tests; Documentation			Debugged Election class output functions - printToAudit(), generateResultsText(), displayResults() to fix issues with STV result outputs., Log invalid ballots while displaying results and audit file: updated the stvballot file's error message - Anwesha Completed System Tests - Anwesha , Unit Tests - Zoe/Anwesha, Documentation - Anwesha
4. Plurality Results Output: As an election official, I want Plurality's election results to print out every candidate's name, so that we can see each candidate and the number of votes they received.	2 hrs	Debug Election class output functions – printToAudit(), generateResultsText() , displayResults() for correct plurality result logging; update pluralityballot error message for invalid ballots; add Unit Tests for regular and tied plurality cases; System Tests; Unit Tests, Documentation			Debugged Election class output functions -> Anwesha/ Zoe : printToAudit(), generateResultsText(), displayResults() for appropriate logging and displaying plurality results, Documentation of Unit Tests - Annabelle, System Tests - Zoe/Anwesha, Documentation - Zoe, Completed Unit Tests: for regular ballots and tied plurality election - Annabelle

5. Audit File Directory: As an election official, I want the audit file to be generated in the /src/ directory, so that the audit.txt file gets stored in the correct directory.	1 hr	Fix path compatibility in printToAudit() (Linux support); complete System Tests; Documentation			Fixed path compability issues in printToAudit() function from Election class, made it Linux compatible - Zoe/Anwesha, System Tests - Zoe, Documentation - Anwesha
6. Shuffle Functionality for STV : As an election official, I want full shuffle functionality in STV elections, so that the STV election output is fair and accurate.	2 hrs	Debug runElection() and setShuffle() in STV class; complete System Testing (ballots before/after shuffle); add Unit Tests in stv_UT.cpp; complete Documentation	Unit tests in stv_UT. cpp, System tests		Debugged runElection() and setShuffle() within STV class - Anwesha, Completed System testing by printing ballots before and after adding shuffle - Anwesha, Completed Documentation - Anwesha, Documentation - Anwesha, System Tests with invalid, mixed ballots and multiple files- Anwesha
7. Implementing MV (Municipal Voting) Algorithm: As an election official, I want to be able to run an MV election algorithm, so that multiple winners can be determined based on the candidates receiving the most ballots with a fair coin toss if there is a tie or ties between candidates.	6 hrs	Code MVlogic as child of Election class, create an mvballot class, add runElection() function within MV class, Refactor Election class accordingly to display results; Unit tests, System tests, Documentation	Refactor UserInterface (getInfo()) & Election (printToAudit(), generateResultsText( ), displayResults(), setBallots()) for MV election logging/display; Documentation	Coding MVlogic class as a child of Election class, refactoring the runElection() function - Hilton, Unit Tests - Zoe, System Tests - Zoe	
8. Obtaining All Election Information from a CSV File (No User Input): As an election official, I want all pertinent election information to be obtained from a file with no user input, so that there can be no user error or possibility of fraud that could cause the election to output the incorrect people/seats.	2 hrs	Refactor getInfo() in UserInterface class & setBallots() in Election class, Refactor Unit Tests for Election class functions [getCSVFileName(), getNumSeats(), setBallots(), getAlgorithm()]; System Tests for STV, MV, Plurality; Documentation		System test for STV, MV, Plurality (taking all inputs from single csv) : Zoe/Anwesha, Documentation of code and unit tests: Annabelle	Refactoring UserInterface class function getInfo() and Election class function setBallots() - Annabelle, Writing Unit tests for Election class and UserInterface class- Annabelle
9. Taking Multiple CSV File Inputs: As an election official, I want to be able to bring in multiple CSV files to the system, so that we can bring in different files from different balloting locations.	4 hrs	Refactor UserInterface to handle multiple CSV inputs – added getCsvFileNames() & modify getInfo() for N inputs; Add Unit Test getCSVFileNameTest in electionunittests; Documentation	System tests with multiple MV files and STV files-Hilton/ Anwesha, Plurality System Test - Zoe, Documentation / Doxygen - Annabelle	Refactoring UserInterface code to take multiple CSV inputs: modify getnfo() to prompt the user to take N no. of input files - Anwesha	

10. STV Ballot Validation & Reporting: As an election official, I want STV ballots with ranking errors to be removed and reported, so that only valid ballots are counted and errors are transparent.	5 hrs	Refactor STV and Election classes for ballot validation and logging – update generateResultsText() and setBallots() in Election, update stvballot.cpp to notify invalid ballots with IDs, Unit Tests, System Tests, Documentation		Refactor STV and Election classes for ballot validation and logging – update generateResultsText() and setBallots() in Election, update stvballot.cpp to notify invalid ballots with IDs , Unit Tests, System Tests, Documentation	
11. Small Election Report: As an election official. I want a small report summarizing election outcomes, so that I can maintain a record of relevant election information.	2 hrs	Refactor generateResultsText() in Election class to add election summaries with valid/invalid counts, vote percentages, winners and losers for MV, STV, and Plurality		Refactor generateResultsText() in Election class to add election summaries with valid/invalid counts, vote percentages, winners and losers for MV, STV, and Plurality	
12. Display MV Election Results: As an election official, I want MV election statistics displayed on-screen after voting, so that I can see winners, losers, and vote counts.	2 hrs	Refactor displayResults() and generateResultsText() in Election class to show MV output on UI , Unit Tests, System Tests, Documentation		Refactor displayResults() and generateResultsText() in Election class to show MV output on UI, Unit Tests, System Tests, Documentation	
	30 hours total				