User Stories	Tasks	Description	Dependency	Story Points	1	2	3	4	5
1	1.1	Write our Tools Specifications that we will be using to build the platform. Documentation of technology we are using. (Sheet Team Plan)		4	R:2	R:2			
1	1.2	Create Heroku project and server.		2		B:2			
1	1.3	Set up Node.js backend on the Heroku server to show a simple web page containing HTML and CSS.	1.2	4			B:2	B:2	
1	1.4	Set up functionality to accept CSV files, by this we mean create the function that will handle the click request		2					B:2
2	2.1	Setup intial layout of web application, which involves node libraries/gitignores and standard files		2	D:2				
2	2.2	Create a simple HTML page with CSS styling (all with Bootstrap) which has a clickable upload button, with the X logo	2.1	10	C:2	C:2	C:2	C:2	C:2
2	2.3	Create a simple HTML page with CSS styled (all with Bootstrap) with Login Page.	2.1	10	A:2	A:2	A:2	A:2	A:2
2	2.4	Create a simple HTML page with CSS styled (all with Bootstrap) with Create Account Page.	2.1	5					
2	2.5	Create a Navigation Bar Sticky Header at the top of the webpage	2.1	5					
3	3.1	Flesh out the System Design of the Database Schema		8		D:2	D:2	D:2	D:2
3	3.2	Code Insert Report EndPoint (Accepts JSON Objects) (No DB Needed)		2	B:2				
3	3.3	Parse the json object element by element and insert it into the database	3.2	14			R:2	R:2	R:2
3	3.4	Use Axios to make a post request to the endpoints to send data.	3.2	8					
3	3.5	Read and parse data from an iCARE template, by using the npm package that converts csv to json	1.4	8					
		PLAN SPRINT # 1							

User Stories	Tasks	Description	Dependency	Story Points	1	2	3	4	5			
1	1.1	Write our Tools Specifications that we will be using to build the platform. Documentation of technology we are using. (Sheet Team Plan)		4	R:2	R:2						
1	1.2	Create Heroku project and server.		2		B:2						
1	1.3	Set up Node.js backend on the Heroku server to show a simple web page containing HTML and CSS.	1.2	4			B:2	B:2				
1	1.4	Set up functionality to accept CSV files, by this we mean create the function that will handle the click request		2					B:0		Byron did not have time to work on his task on day 5	
2	2.1	Setup intial layout of web application, which involves node libraries/gitignores and standard files		2	D:2							
2	2.2	Create a simple HTML page with CSS styling (all with Bootstrap) which has a clickable upload button, with the X logo	2.1	10	C:2	C:2	C:2				D decided to work on 2.4 first, and finished it on the first day in 2 hours ... so he worked on the database schema on days 3 4 5, and was able to finish it in a total of 6 developer hours, he also finished 3.1 on itme	
2	2.3	Create a simple HTML page with CSS styled (all with Bootstrap) with Login Page.	2.1	10	A:2	A:2	A:2	A:2	A:2			
2	2.4	Create a simple HTML page with CSS styled (all with Bootstrap) with Create Account Page.	2.1	5		D:2						
2	2.5	Create a Navigation Bar Sticky Header at the top of the webpage	2.1	5				C:2	C:2			
3	3.1	Flesh out the System Design of the Database Schema		8			D:2	D:2	D:2		C finished his 2.2 task earlier than expected, and so worked on the 2.5 task, but did not end of completing it	
3	3.2	Code Insert Report EndPoint (Accepts JSON Objects) (No DB Needed)		2	B:2							
3	3.3	Parse the json object element by element and insert it into the database	3.2	14			R:2	R:2	R:2			
3	3.4	Use Axios to make a post request to the endpoints to send data.	3.2	8								
3	3.5	Read and parse data from an iCARE template, by using the npm package that converts csv to json	1.4	10								
		EXECUTION SPRINT # 1										

Days	0	1	2	3	4	5
Provisial (in story points)	50	46	40	40	36	6
Actual (in story points)	**50**	46	35	25	21	3
BURNDOWN SPRINT # 1						

User Stories	Tasks	Description	Dependency	Story Points	1	2	3	4	5
1	1.4	Set up functionality to accept CSV files, by this we mean create the function that will handle the click request (be able to upload)		2	B:2				
2	2.5	Create a Navigation Bar Sticky Header at the top of the webpage		1	C:1				
2	2.6	Create the landing page and hook it up to the navbar		3	C:1	C:2			
2	2.7	Create the about us page and hook it up to the navbar		4	A:2	A:2			
2	2.8	Create a page that contains history of events that the TEQ can see. Make the UI only. In future we will get the data from the database.		4				C:2	C:2
2	2.9	Create a logout button for the page that the organization member goes to after they login, the local storage should reflect that they log out		2			C:2		
2	2.10	Create a Grid using an NPM Package for the list of people in the organization		6			A:2	A:2	A:2
3	3.3	Parse the json object element by element and insert it into the database		8	R:0	R:2	R:2	R:2	R:2
3	3.4	Read and parse data from an iCARE template, by using the npm package that converts csv to json	1.4	8		B:2	B:2	B:2	B:2
3	3.5	Send Axios to make post request when logging into account with local storage setup		2	D:2				
3	3.6	Send Axios to make post request when creating organization account		2		D:2			
3	3.7	Send Axios to make post request when uploading file data		6			D:2	D:2	D:2
		PLAN SPRINT # 2							

User Stories	Tasks	Description	Dependency	Story Points	1	2	3	4	5		
1	1.4	Set up functionality to accept CSV files, by this we mean create the function that will handle the click request (be able to upload)		2			B:2				C73 assignment, midterms
2	2.5	Create a Navigation Bar Sticky Header at the top of the webpage		1	C:1						
2	2.6	Create the landing page and hook it up to the navbar		3	C:1	**C:1**					
2	2.7	Create the about us page and hook it up to the navbar		4		**C:1**					
2	2.8	Create a page that contains history of events that the TEQ can see. Make the UI only. In future we will get the data from the database.		4				C:2	C:2		
2	2.9	Create a logout button for the page that the organization member goes to after they login, the local storage should reflect that they log out		2			C:2				
2	2.10	Create a Grid using an NPM Package for the list of people in the organization		6							Was not done, low priority
3	3.3	Parse the json object element by element and insert it into the database		8	R:0	R:2	R:2	R:2	R:2		
3	3.4	Read and parse data from an iCARE template, by using the npm package that converts csv to json	1.4	8				B:4	B:4		
3	3.5	Send Axios to make post request when logging into account with local storage setup		2	D:2						
3	3.6	Send Axios to make post request when creating organization account		2		D:2					
3	3.7	Send Axios to make post request when uploading file data		6			D:2	D:2	D:2		
		EXECUTION SPRINT # 2									

Days	0	1	2	3	4	5
Provisial (in story points)	50	45	36	34	34	2
Actual (in story points)	**50**	47	38	34	34	8
BURNDOWN SPRINT # 2						

	React + SC	JavaScript + CSS + HTML	
Aliza	Styled Components	CSS	
Bryon	JavaScript Front-End	Javascript	
Caleb	Styled Components	CSS	
Dharmik	JavaScript Front-End	JavaScript	
Ralph	BackEnd + DataBase (Node Express and NEDB/MongoDB)	Node/Express/MongoDB	
Sprint #1	Sprint #2		
Wednesday - Sunday	Mon - Fri		
Git Tutorial			
Git Version Control	https://try.github.io/		
How to Build & Style a Web Page		Caleb and Aliza google how to format a PDF with CSS	
BootStrap (CSS Styling)	https://getbootstrap.com/		
Using HTML and CSS, bootstrap	https://www.w3schools.com/html/html_css.asp		
Jquery (Making Web Dev Faster)	https://jquery.com/		
How to talk to back-end from FrontEnd			
JSON OBJECTS (How data Goes over the web)	https://www.json.org/		
Fetch or Axios Networking Libarary for Frontend	https://www.npmjs.com/package/axios		
How backend receives information from Front End			
Node.js	https://nodejs.org/en/		
Express.js	https://expressjs.com/	app.get('/', (req, res) => res.send('Hello World!'))	This endpoint is what the front end people will hit. So LocalHost:/ will return Hello World when hit with a GET request
How to Store Data			
NEDB DataBase Technology	https://github.com/louischatriot/nedb		
Where will our code be deployed			
Heroku it's like a Web Server	https://www.heroku.com/	When you push to master you will see updates at the URL Dharmik's Website.com	Ralph should set up CD
Packages			
Grid Package	https://gijgo.com/grid/demos/bootstrap-4-table	https://www.npmjs.com/package/jquery-grid	
Upload a CSV file, return a JavaScript Array	https://www.npmjs.com/package/csvtojson	This converts CSV Files to JSON Objects that we can send to the Server Automatically since it's a JSON Object	
	*/** csv file*	Converts to JSON Object	
	a,b,c	*[*	
	1,2,3	** {a:"1", b:"2", c:"3"},*	
	4,5,6	** {a:"4", b:"5". c:"6"}*	
	**/*	**]*	
Helpful Links			
Bryon's Task	https://stackoverflow.com/questions/4720343/loading-basic-html-in-node-js		