

ID	Story	Estimation (hours)	Priority (1-5)	Sprint When Finished
1	Set up a PostgreSQL database on AWS using Docker	3	3	1
2	Design and implement the database schema for inventory	4	4	1
4	Set up a Node.js/Express.js project structure	3	1	1
5	Implement user authentication in the API	5	5	1
7	Set up a React project structure for the front-end	3	1	1
8	Implement login and registration pages in React	5	4	1
9	Connect the React front-end to the Node.js API for authentication	4	2	1
16	Figure out best way to convert invoice info	2	1	1
22	Update Database Schema	2	5	1
23	Create API endpoints for login	8	5	1
24	Collapsible sidebar on all pages	4	3	1
25	Logout setup	3	4	1
3	Populate the database with sample inventory data	2	1	2
6	Create API endpoints for inventory management	8	1	2
10	Design the inventory management UI in React	6	2	2
11	Implement inventory listing in the React app	4	2	2
12	Implement adding new inventory items in the React app	5	2	2
19	Design the dashboard for the app	5	2	2
27	Fix project crashing on refresh	3	3	2
28	Password security - 8 chars, special char, numbers, capital	2	5	2
30	Create Jobs page frontend	4	5	2
31	Fix Logout to guarantee modal operation	1	2	2
32	Add out of stock/instock in inventory page	2	3	2
33	Fix top bar stutter	3	5	2
34	Add change quantity implementation	3	3	2
36	Implement checkbox filtering	2	5	2
40	Remove duplicates in filter	1	2	2
41	Added charts to the dash	3	1	2
43	Add Jobs table to database	1	4	2
44	Create a route for the jobs page	1	4	2
48	Create a filter for the Jobs page	2	3	2
49	Connecte the add jobs button with the API	3	4	2
51	Create the title for the Frontend of the jobs page	0.5	4	2
52	Implement a dropdown function for each job	1	4	2
53	Setup a Reverse Proxy	4	2	2
54	Created Self-Signed SSL certificates	0.5	2	2
55	Add put, get, post, patch to products api	4	4	2
56	Updated the REST API to use HTTPS instead of HTTP	2	2	2
57	Updated the React App to use HTTPS instead of HTTP	1	2	2
59	Fix the JWT in the Front End to kick a user once the JWT has expired	1.5	3	2
60	Allow users to edit information such as username, password, and email	5	4	2
62	Add Purchases table to database	6	4	2
63	Create a route for the purchases page	1	3	2

66	Create a filter for the purchases page	1.5	3	2
13	Implement inventory warnings and notifications	4	1	
14	Deploy the Node.js API on AWS using Docker	4	1	
15	Deploy the React front-end on AWS using Docker	4	1	
17	Compute average based on past invoices	3	1	3
18	Design the inventory management page	5	3	3
20	Design the invoice management page	5	3	3
21	Scanner for the invoices to automatically insert data	5	2	3
26	Set up a setting page	5	2	3
29	Remember me	2	5	
35	Add remove item in inventory	4	4	3
37	Implement inventory search feature	3	2	3
38	Investigate the server that will hold the app	5	5	3
39	Communicate with Server IT to get access	5	5	3
42	Add Modal to input orders on order details page	2	3	3
45	Implement edit button for jobs page	2	3	3
46	Implement delete button for jobs	2	3	3
47	Create a table for estimates and figure out calculation	2	3	3
50	Remove the invoice section on the jobs page	1	3	3
58	Fix editing the partnumber of a product	2	4	3
61	Learn how to put an app on a server	4	5	3
64	Implement edit button for purchases page	2	4	3
65	Implement delete button for purchases	3	4	3
67	Implement a lockout feature	2	4	3
68	Add functionality to edit necessary parts	2	4	3
69	Add functionality to edit used parts	2	4	3
70	Add a modal to be able to add parts to necessary	3	5	3
71	Add functionality to remove a necessary part	2	4	3
72	Add functionality to the remove used parts button	2	4	3
73	Add functionality to the move to necessary button	3	4	3
74	Create totals of the prices for Used and necessary parts	1.5	4	3
75	Edit the UI so the Estimate, necessary parts, and used parts sections are stacked	1	4	3
76	Edit the Estimate UI	0.5	4	3
77	Create a scraper for the Estimates page	5	2	3
78	Add a modal to be able to add parts to used + functional	2	4	3
79	Calculate the total prices of the parts in necessary + used	1	3	3
80	Jobs page filter should be filtering through database instead of UI	2	1	3
81	Limit the upload size of a PDF in the estimates page.	1	2	3
82	DNS setup for the server	1	3	3
83	Hosting project on a server	4	5	3
84	Set up a production version of the Docker Compose file for the server	3	4	3
85	Create a modal for jobs Remove function	2	2	3
86	Create a modal for jobs move to used function	2	5	3
87	Create a modal for jobs move to necessary function	2	5	3

88	Create a modal for jobs remove from used	2	4	3
89	Create a modal for jobs remove from necessary	2	4	3
90	Add estimate file name in estimate box	1	5	3
91	Implement logging page for all features in the application	3	4	3
92	log actions when adding product in inventory	1	3	3
93	log actions when removing product in inventory	1	2	3
94	log actions fro editing product details in inventory	1	2	3
95	log actions for editing product instock quantity in inventory	1	3	3
96	Fix topbar implementation on mobile	4	5	3
97	Fix the mobile look for inventory	1	3	3
98	Remove checkboxes and filter headers from mobile look	1	3	3
99	Adjust inventory table sizing and display on mobile	3	3	3
100	Fix logging for inventory	2	5	3
101	Implement logging for adding jobs	6	3	3
102	Setup GitHub action to automatically run Cypress Test whenever a pull request is made to the main branch	2	4	3
103	Setup GitHub action to automatically update server whenever merge is made to the main branch	1	5	3
104	Log actions when adding a job	1	5	3
105	Log actions when adding an estimate to a job	1	5	3
106	Log actions when adding a necessary part	1	5	3
107	Log actions when adding a used part	1	5	3
108	Log actions when moving to used	1	5	3
109	Log actions when moving to necessary	1	5	3
110	Log actions when removing part	1	5	3
111	Log actions when editing quantity needed	1	5	3
112	Jobs - Create modals for all functions	7	5	3
113	Make the dashboard based on live data in database	3	1	3
114	Change graphical data to tabled data for product stock dashboard	3	2	3
115	Create initial setup modal for first user	4	4	3
116	Move initial setup modal to login	2	4	3
117	Remove autoregister first user	2	4	3
118	Remove Register route in App.js	1	2	3
119	Implement Role Based Access Control	6	5	3
120	Move register new user to dropdown in topbar	3	3	3
121	Implement manage users to change permissions of users or remove them	4	5	3
122	Implement check to always have 1 admin in the app	2	4	3
123	Hide logs from people who are not role admin	3	5	3
124	Add modals to Purchases Page	2	2	3
125	Create pdf scraper for Purchase orders	6	1	
126	Email verification	2	1	
127	Access in house Guaranty metal server (waiting on server installation)	8	1	
128	Put app on in house guaranty metal server (waiting on server installation)	6	1	
129	Update json web token for editing profile username	2	4	3
130	Update json web token for editing profile password	2	4	3
131	Update json web token for editing profile role	2	4	3
132	Update Database to reflect new purchases page tables	2	5	3

133	Rebuild data structure to work with new tables on purchases	2	5	3
134	Implement api's for moving items into a new order from low and out of stock	7	5	3
135	Implement front end purchases page	8	5	3
136	Implement status logic for purchases	3	5	3
137	Implement inventory updates based off of status logic	3	5	3
138	Implement mark up price to use shipping costs	3	4	3
139	Create the generate xlsx button for purchases	4	4	3
140	Implement delete orders function with confirmation	3	3	3
141	Update modals for purchases	2	3	3
142	Add Testing for the Jobs page on Cypress	4	5	3
143	Edit the upload estimate button so you cannot upload if there is no file chosen	1	4	3
144	Add logging for purchases	4	4	3
145	Upload invoice pdf implementation	2	2	
146				
147				
148				
149				
150				
151				
152				
153				
154				
155				
156				
157				
158				
159				
160				
161				
162				
163				
164				
165				
166				
167				
168				
169				
170				
171				
172				
173				
174				
175				
176				
177				

178				
179				
180				
181				
182				
183				
184				
185				
186				
187				
188				
189				
190				
191				
192				
193				
194				
195				
196				
197				
198				
199				
200				
201				
202				
203				
204				
205				
206				
207				
208				
209				
210				
211				
212				
213				
214				
215				
216				
217				
218				
219				
220				
221				
222				

223				
224				
225				
226				
227				
228				
229				