

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	3	
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	
18	Design the inventory management page	5	3	
20	Design the invoice management page	5	3	
21	Scanner for the invoices to automatically insert data	5	2	
26	Set up a setting page	5	2	
29	Remember me	2	5	
35	Add remove item in inventory	4	4	
37	Implement inventory search feature	3	2	
38	Investigate the server that will hold the app	5	5	
39	Communicate with Server IT to get access	5	5	
42	Add Modal to input orders on order details page	2	3	
45	Implement edit button for jobs page	2	3	
46	Implement delete button for jobs	2	3	
47	Create a table for estimates and figure out calculation	2	3	
50	Remove the invoice section on the jobs page	1.5	3	
58	Fix editing the partnumber of a product	2	4	
61	Learn how to put an app on a server	2	5	
64	Implement edit button for purchases page	2	4	
65	Implement delete button for purchases	3	4	
67				
68				
69				
70				
71				
72				
73				
74				
75				
76				
77				
78				
79				
80				
81				
82				
83				
84				

85				
86				
87				
88				
89				
90				
91				
92				
93				
94				
95				
96				
97				
98				
99				
100				
101				
102				
103				
104				
105				
106				
107				
108				
109				
110				
111				
112				
113				
114				
115				
116				
117				
118				
119				
120				
121				
122				
123				
124				
125				
126				
127				

128				
129				
130				
131				
132				
133				
134				
135				
136				
137				
138				
139				
140				
141				
142				
143				
144				
145				
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				