Software testing is a process of evaluating a software solution to ensure that it behaves as expected and meets the specified requirements.

Testing strategy

After our first print planning for sprint one, we decided that we would follow a test-driven development approach, where we write tests first to make sure that very function, class or routes created follow what is expected. But due to time constraints we would use a different approach.

For each piece of functionality that is built and completed, we would write tests for that to ensure it works as it should do that when other functionalities are added to that one, they should not fail the previously tested functionalities.

- We used **white-box testing** to deep tests the structure of the code and if database implementations along with our api functions as expected and ensure it stays like that.
- For the front-end part or specifically the UI we used **black-box testing** by ensuring that we test every element to see if it works properly and if it will be visually appealing and user friendly.

Test cases and results

By following this strategy we were able to do upto 70% test coverage, which we were struggling with in our first sprint since we needed to change a lot of things when everything started. We managed to achieve the following:

- Easy to detect defects in the code
- Ensure that we deliver a quality softare as best as possible
- Improve user experience
- Improved collaboration amongst ourselves as developers

Test Coverage Results

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	70.29	80	60	70.29	
api/mongoDB	100	100	100	100	
...pplication.js	100	100	100	100	
...er_Project.js	100	100	100	100	
Project.js	100	100	100	100	
User.js	100	100	100	100	
description.js	100	100	100	100	
controller	62.96	76.92	60	62.96	
controller.js	62.96	76.92	60	62.96	...28,137,145-147