

## Possible enhancements

### From the **functional testing** standpoint

I feel the need for **automating the regression tests** with all the major end-to-end scenarios of this app.

The **regression suite can be linked to a Jenkins job** which will run each time a developer pushes changes. This will help in Continuous integration and development.

Other Automated tools that can be used:

As the app is developed in Ruby on Rails and Angular, we could use **RSpec with capybara or Protractor using javascript**.

If you are using Rails, put Capybara specs in spec/features and Capybara specs in a different directory, then tag the example groups with :type => :feature.

Protractor is an end-to-end test framework for Angular and AngularJS applications.

Protractor needs two files to run, a spec file and a configuration file.

### For non-functional testing

The To-do App needs to go through some **performance and scale testing**

Automating tools like **JMeter and LoadRunner** can be used for this.

Also, some web API testing should be done to check the data returned from **API endpoints**.

API testing tools like **Postman or RunScope** could be used.

To maximize the value of testing,

I design a **test coverage checklist and exploratory tests** in addition to the acceptance criteria's  
Eg:

As a ToDo App user

I should be able to create a task

So I can manage my tasks

### Test Coverage checklist

- ~~The user should always see the 'My Tasks' link on the NavBar~~ ○ Clicking this link will ~~redirect the user to a page with all the created tasks so far~~ - pass
- The user should see a message on the top part saying that list belongs to the logged user-  
**Fail – The displayed message is “John’s ToDo List”**
- ~~The user should be able to enter a new task by hitting enter or clicking on the add task button.~~ - pass
- The user should not be able to add an empty task by clicking on “+” add symbol -**fail**
- The task should require at least three characters so the user can enter it. – **Fail – able to add tasks with less than three characters**

- The task can't have more than 250 characters. – Fail – able to add a task with 254 characters
- When I add a very long name for the task it shouldn't break the page – fail – text overflow issue
- Task with same name shouldn't exist – Fail – able to add tasks with same name
- When added, the task should be appended on the list of created tasks –fail – when a new task is added it appears as the first task on the list of created tasks

### Exploratory tests

Test in different browsers and OS, make use of selenium grid for parallel testing.