Appendix A - Testing

When beginning the project, the team had an initial meeting and discussed the functional requirements that the project must satisfy. These requirements were outlined in the Module Specification Document (accessible via this link: <https://drive.google.com/file/d/1699OIMWQZYco6nCQE6cMLdUDSyRtTQ9C/view?usp=sharing> ). From this document, a list of functional requirements were identified for the application which are displayed below.

The application will need:

* A login and registration system
* Appropriate hashing of passwords
* Milestones to be stored in a database
* Milestones to have:
  + Name
  + Description
  + Due Date
  + Actual Completion Date that reflects when the date of the milestone is completed
* CRUD setup for milestones
* Functionality to show a list of incomplete milestones
* Functionality to share milestones (via link)

We analysed the functional requirements understood that simply put; the application must allow users to create milestones to track activities. Using our gathered knowledge of milestone planners, we decided to expand on these requirements by adding a projects class in the system as in industry milestones are categorised into projects. Below are our additional requirements we wanted to satisfy:

The application will need:

* To be able to create projects, which in turn must have the following attributes:
  + Name
  + Start Time
  + End Time
  + Description

In retrospective, we believe that this was a good decisive action as this allowed the user to group their milestones as is done in industry. As confirmed in the testing conducted below, we satisfied all functional requirements and additional functional requirements outlined by the group. For our testing methodology, we decided to divide the testing in to subtopics to ensure testing was conducted for different aspects of the application. Below are the subsections of testing:

1. User Interface Display Testing
2. Input Validation Testing
3. Responsiveness Testing
4. User Access Testing

White-box testing will be used to ensure all requirements within the specification have been met. A table will be used with the headings of Test ID, Test Name, Test Type, Expected Result, Actual Result and Screenshot. Normal testing will be done on all functionality, Extreme and exceptional testing will only be carried out where necessary and allows for. We believe that this process has been a tried and tested process in previous projects and has yielded an accurate representation of the system. As shown below, it is an accurate manner of ensuring all functionality is tested.

# Application Testing

## 1 - User Interface Display Testing

User interface display testing is the process of ensuring all displays on the application work according to the aims of the team. This will be tested on a desktop computer using Google Chrome as the browser. The ‘Test Name’ category has been replaced with the ‘View Name’ as the display of the view will be tested.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Name** | **Test Type** | **Expected Result** | **Actual Result** | **Screenshot** |
| 1 | Initial Interface | Normal | The view will render on the screen. | As Expected | 1.1 |
| 2 | Initial Interface when logged in | Normal | The view will render on the screen. | As Expected | 1.2 |
| 3 | Register Interface | Normal | The view will render on the screen. | As Expected | 1.3 |
| 4 | Log In Interface | Normal | The view will render on the screen. | As Expected | 1.4 |
| 5 | My Projects Interface (unpopulated) | Normal | The view will render on the screen. | As Expected | 1.5 |
| 6 | My Projects Interface (populated) | Normal | The view will render on the screen. | As Expected | 1.6 |
| 7 | Exemplar Milestones Interface (unpopulated) | Normal | The view will render on the screen. | As Expected | 1.7 |
| 8 | Exemplar Milestones Interface (populated) | Normal | The view will render on the screen. | As Expected | 1.8 |
| 9 | Exemplar Shared Milestone | Normal | The view will render on the screen. | As Expected | 1.9 |

## 2 - Input Validation Testing

Input validation testing is the process of ensuring all input boxes that the user can write information into on the application work according to the aims of the team. This will be tested on a desktop computer using Google Chrome as the browser and will have Normal, Extreme and Exceptional testing to ensure that the system is robust.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Name** | **Test Type** | **Expected Result** | **Actual Result** | **Screenshot** |
| 1 | User enters registration details | Normal | User enters valid credentials and account will successfully be created and logged in. | As Expected | 2.1 |
| 2 | User enters registration details | Extreme | User enters email address that has been previously used to create an account, or the password does not contain a special character, number or uppercase letter, an error message is displayed “There was an error registering you ” | As Expected | 2.2 |
| 3 | User enters login details | Normal | The user enters valid login details, they are then transported to the “My Projects” page | As Expected | 2.3 |
| 4 | User enters login details | Extreme | The user has not yet created an account but tries to login, error message to display “Email has not yet been registered, please try registering first” | As Expected | 2.4 |
| 5 | User Adds a project | Normal | User adds project with start and end date, and name. Project is added to the my projects page | As Expected | 2.5 |
| 6 | user adds a project | Normal | User adds a project starting before the current date. Project is added to my projects page | As Expected | 2.6 |
| 7 | User adds a project | Normal | User adds a project that starts and finishes before the current date. Project is added to my projects page | AS Expected | 2.7 |
| 8 | User adds a milestone | Normal | User Selects a project to view its milestone, clicks to add milestone. Name, description and due date. Successfully added milestone, and returned to the milestone page | As Expected | 2.8 |
| 9 | User edits a milestone | Normal | User clicks the edit milestone button, and changes the due date of the milestone, the milestone is successful updated and the user is returned to the milestone page | As Expected | 2.9 |
| 10 | User deletes a milestone | Normal | User clicks to delete the milestone button, the milestone is successfully deleted | As Expected | 2.10 |
| 11 | User completes milestone | Normal | User clicks to complete milestone, the milestone is then moved to the right of the screen under the complete heading. | As Expected | 2.11 |
| 12 | User shares a milestone | Normal | User clicks to share the milestone, a sharable link is created to copy to other people | As Expected | 2.12 |
| 13 | User edits a project | Normal | User clicks to edit project, and changes details such as description, the changes are made and the my project page is displayed | As Expected | 2.13 |
| 14 | User deletes project | Normal | User clicks to delete project, the project is removed from my project page | As Expected | 2.14 |

## 

## 3 - Responsiveness Testing

Responsiveness testing is the process of ensuring all displays on the application work on a small smartphone device according to the aims of the team. This will be tested on a desktop computer with a smartphone emulator in Google Chrome.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Name** | **Test Type** | **Expected Result** | **Actual Result** | **Screenshot** |
| 1 | Initial Interface | Normal | The view will render on the mobile screen. | As Expected | 3.1 |
| 2 | Initial Interface when logged in | Normal | The view will render on the mobile screen. | As Expected | 3.2 |
| 3 | Register Interface | Normal | The view will render on the mobile screen. | As Expected | 3.3 |
| 4 | Log In Interface | Normal | The view will render on the mobile screen. | As Expected | 3.4 |
| 5 | My Projects Interface (unpopulated) | Normal | The view will render on the mobile screen. | As Expected | 3.5 |
| 6 | My Projects Interface (populated) | Normal | The view will render on the mobile screen. | As Expected | 3.6 |
| 7 | Exemplar Milestones Interface (unpopulated) | Normal | The view will render on the mobile screen. | As Expected | 3.7 |
| 8 | Exemplar Milestones Interface (populated) | Normal | The view will render on the mobile screen. | As Expected | 3.8 |
| 9 | Exemplar Shared Milestone | Normal | The view will render on the mobile screen. | As Expected | 3.9 |

## 

## 4 - User Access Testing

User access testing is the process of ensuring a user that is not logged in cannot get access to users’ content. This will be tested on a desktop computer using Google Chrome.

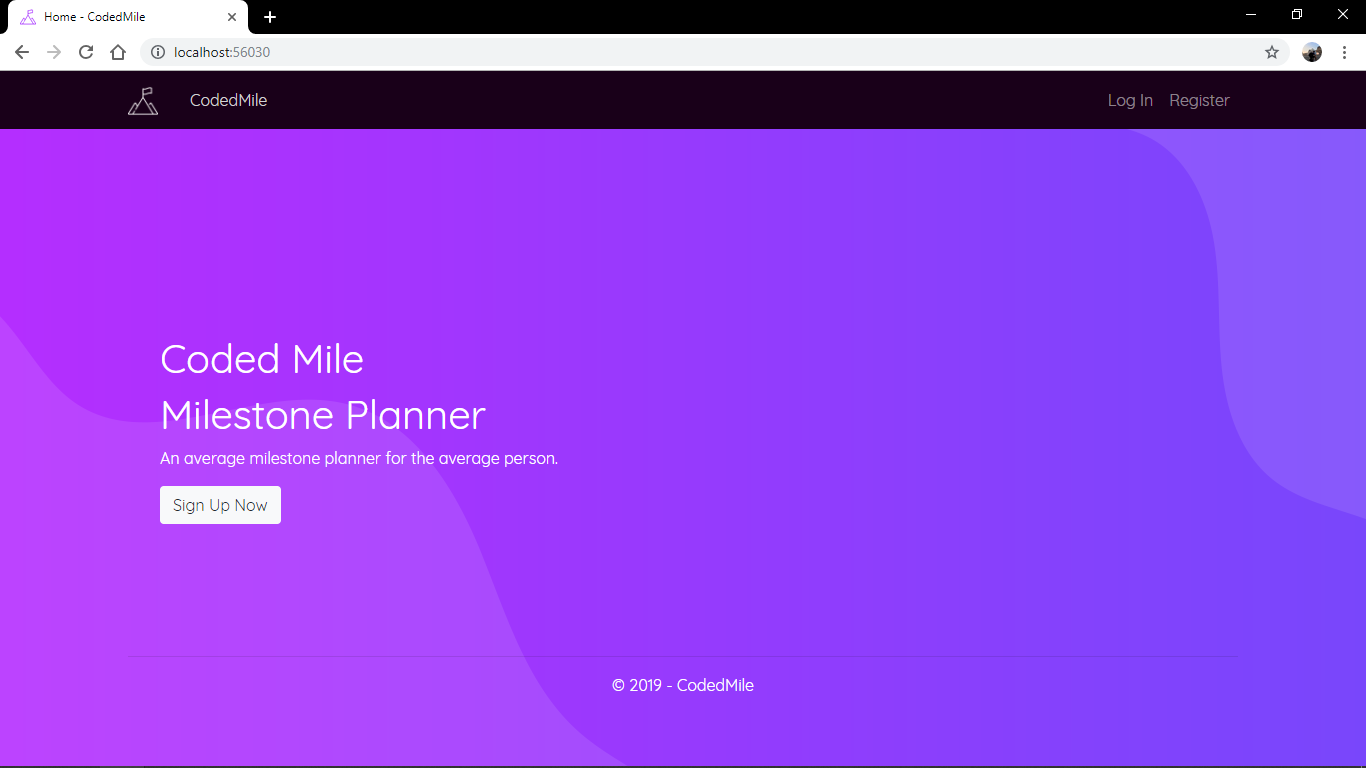
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Name** | **Test Type** | **Expected Result** | **Actual Result** | **Screenshot** |
| 1 | User without account tries to access the Initial Interface | Normal | The user is able to access the view. | As Expected | 4.1 |
| 2 | User without account tries to access the Register Interface | Normal | The user is able to access the view. | As Expected | 4.2 |
| 3 | User without account tries to access the Log In Interface | Normal | The user is able to access the view. | As Expected | 4.3 |
| 4 | User without account tries to access the My Projects Interface | Normal | The user is not able to access the view and is redirected to the login page. | As Expected | 4.4 |
| 5 | User without account tries to access the Exemplar Milestones Interface | Normal | The user is not able to access the view and is redirected to the login page. | As Expected | 4.5 |
| 6 | User without account tries to access the Exemplar Shared Milestone | Normal | The user is able to access the view. | As Expected | 4.6 |

# 

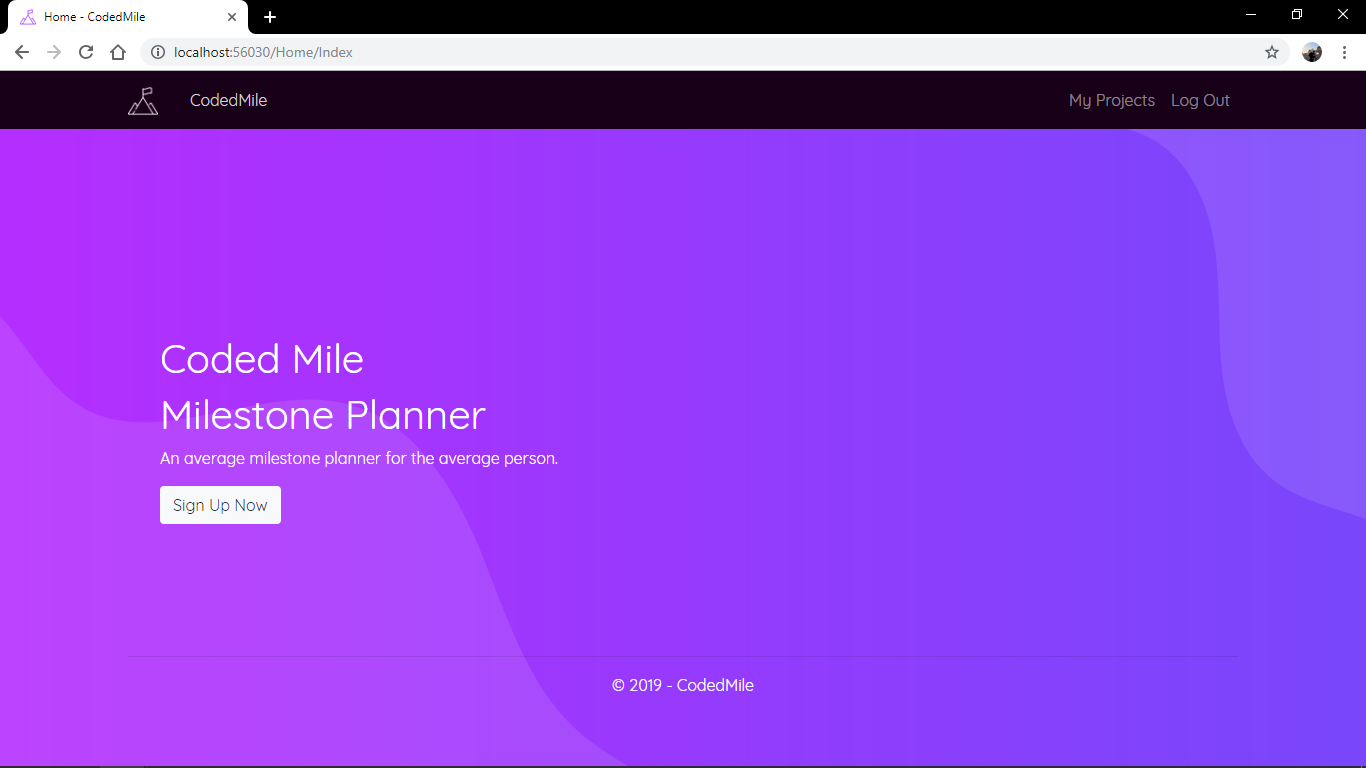
# Testing Screenshots

## 1 - User Interface Display Testing

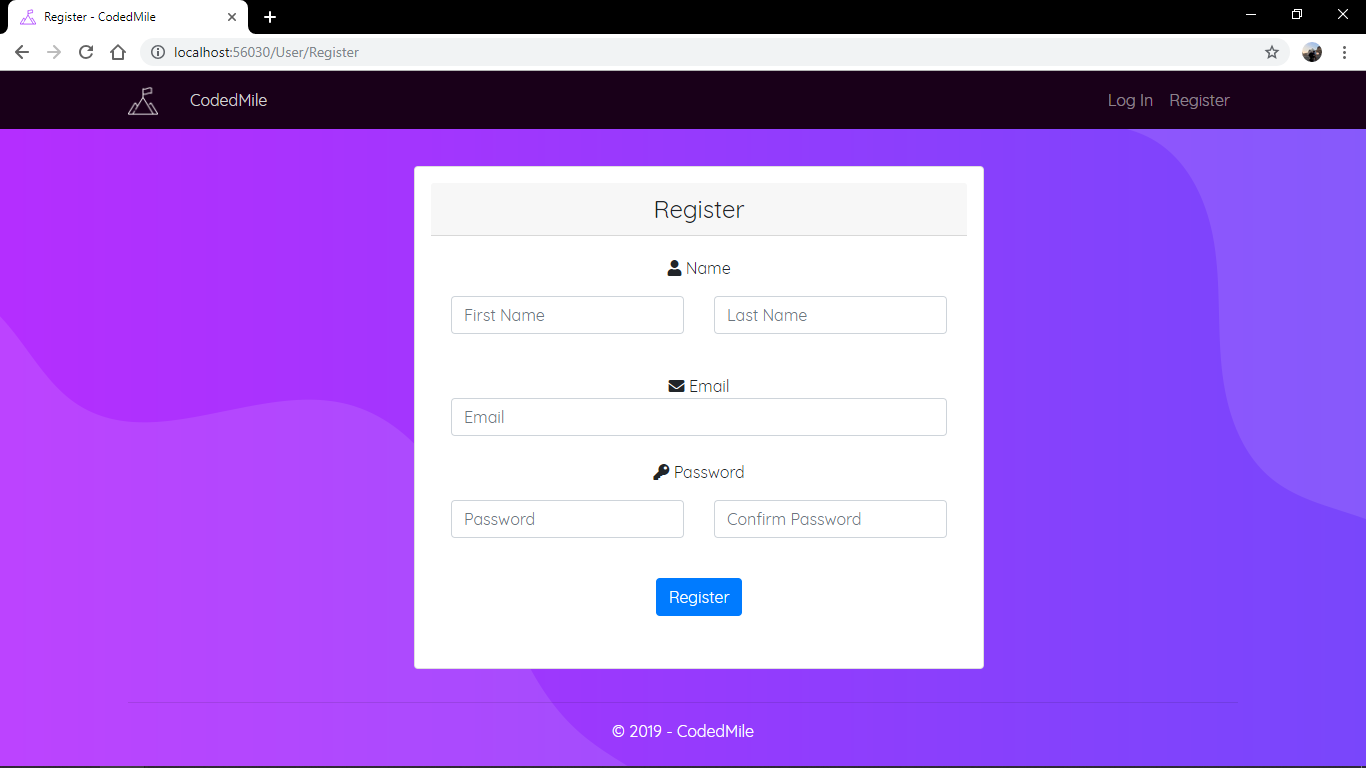
1.1)



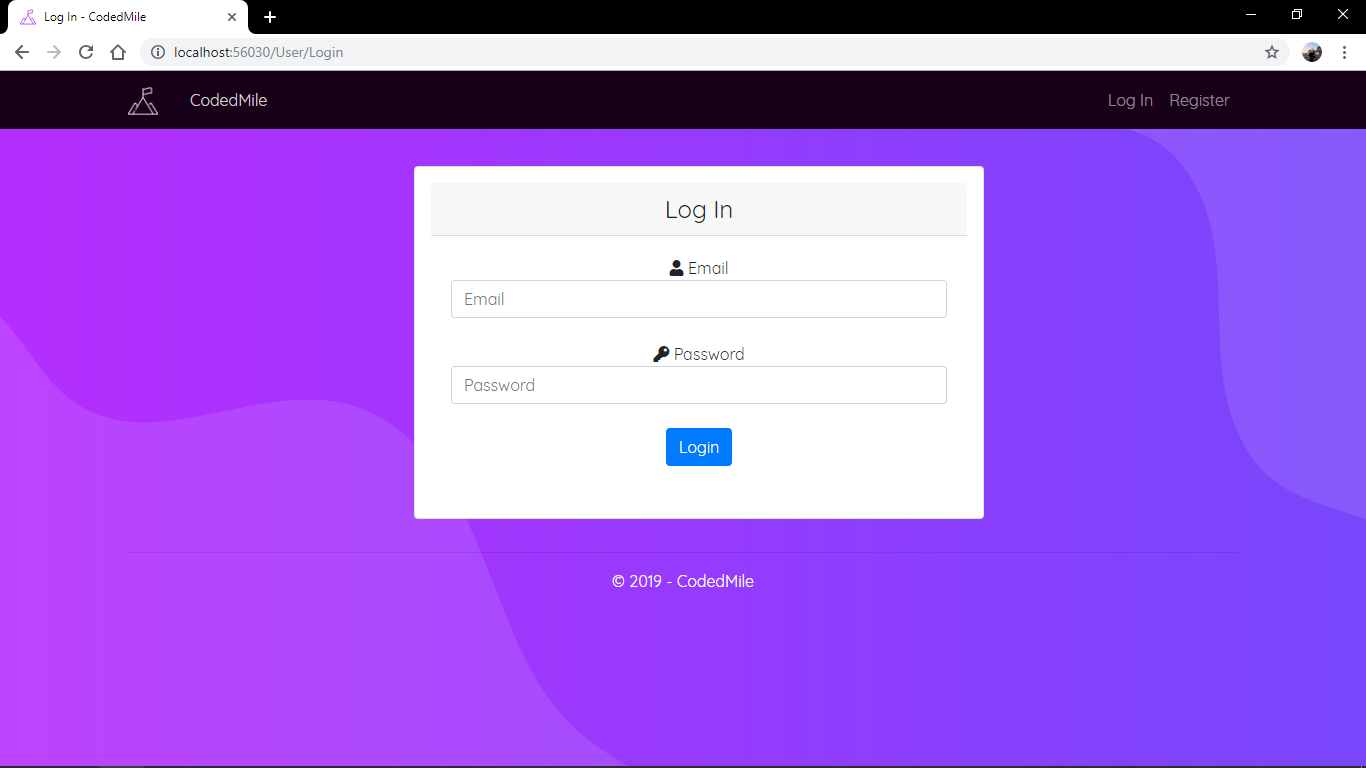
1.2)



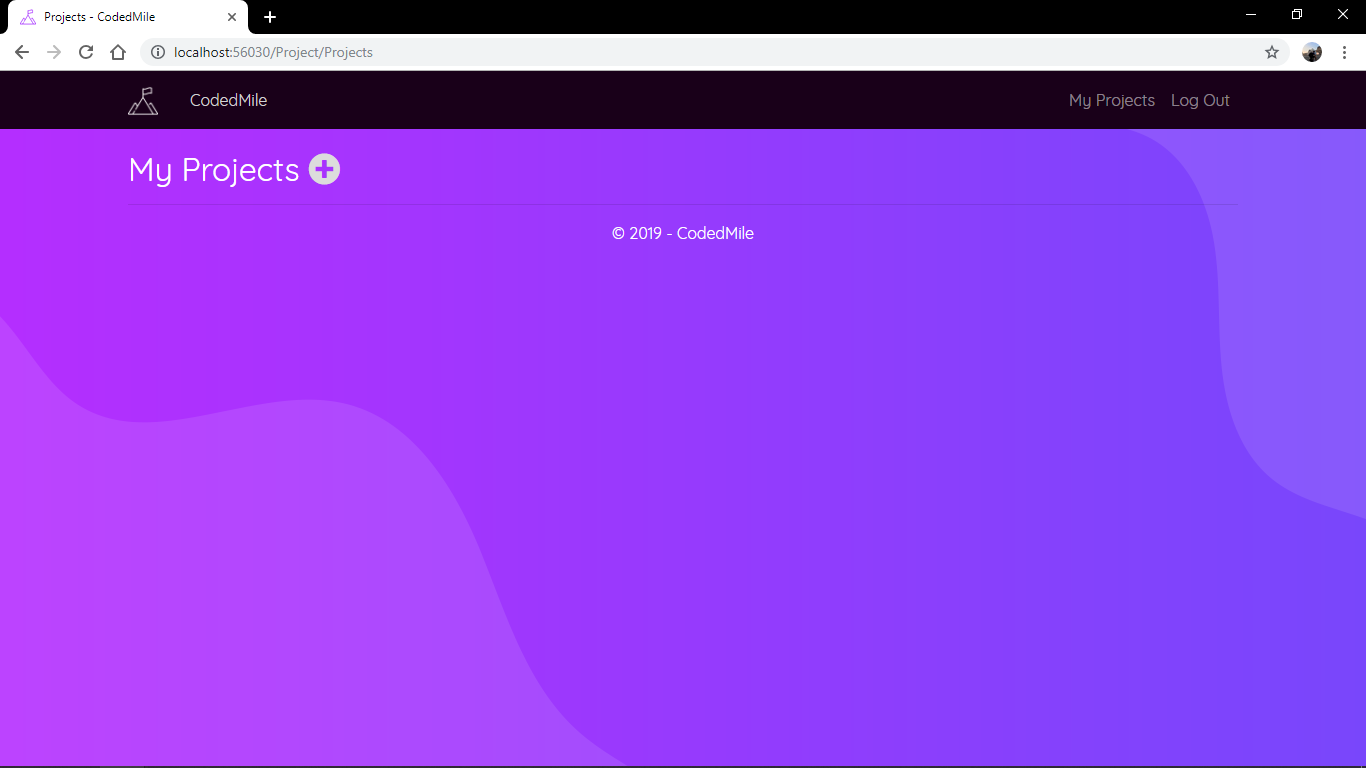
1.3)



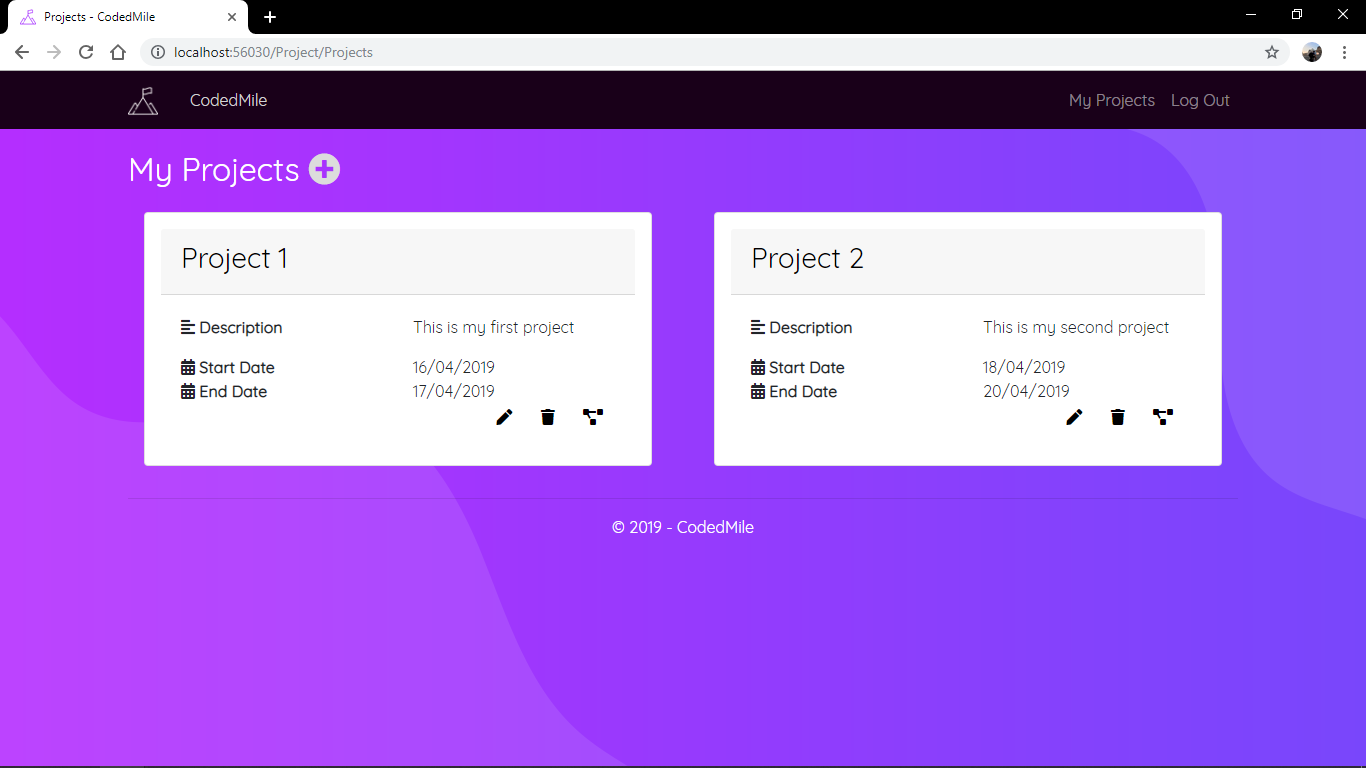
1.4)



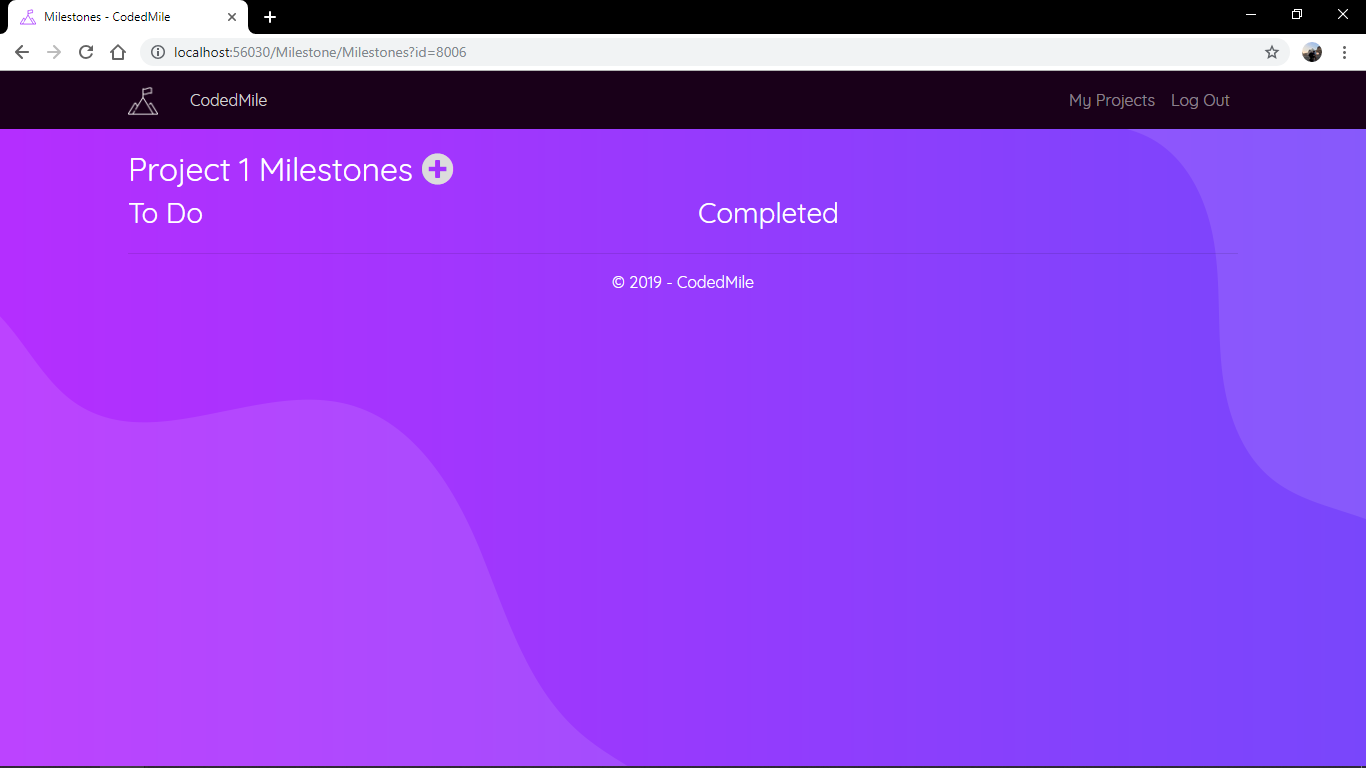
1.5)



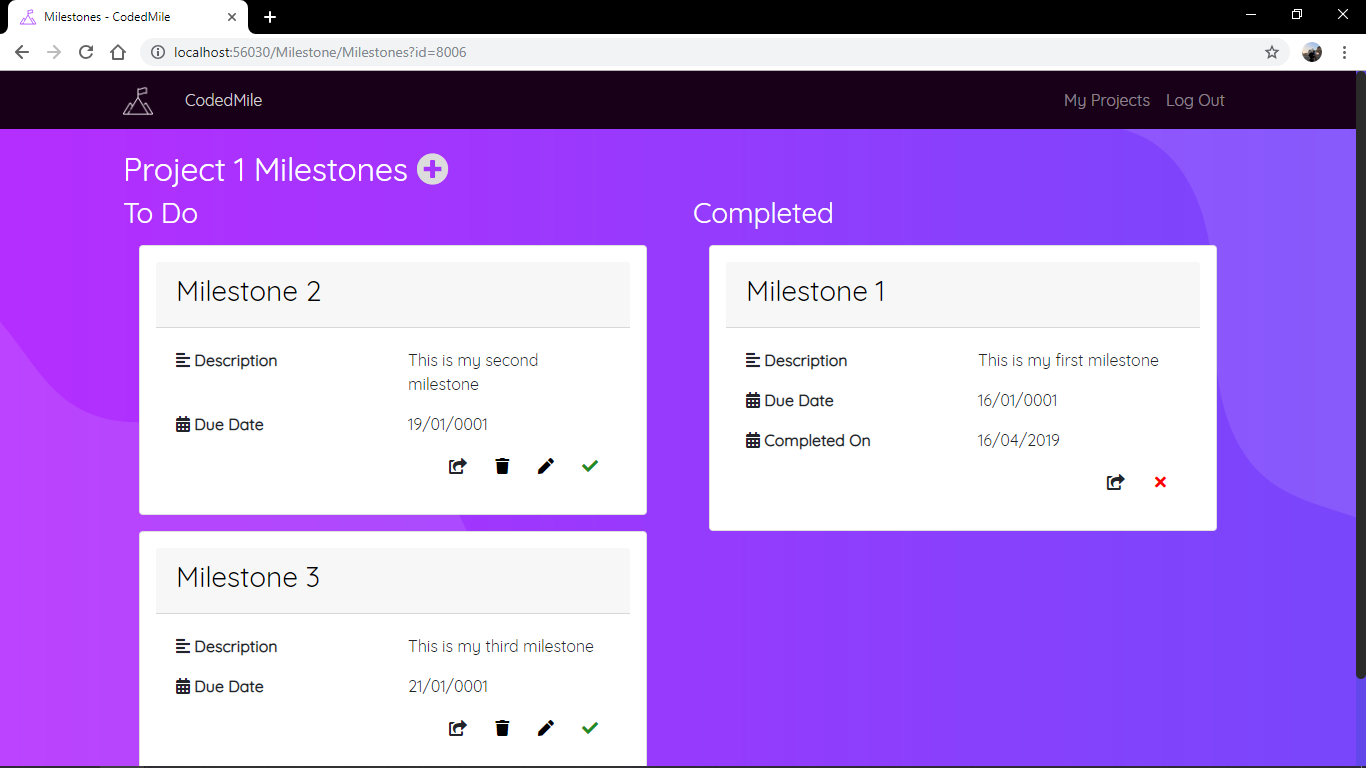
1.6)



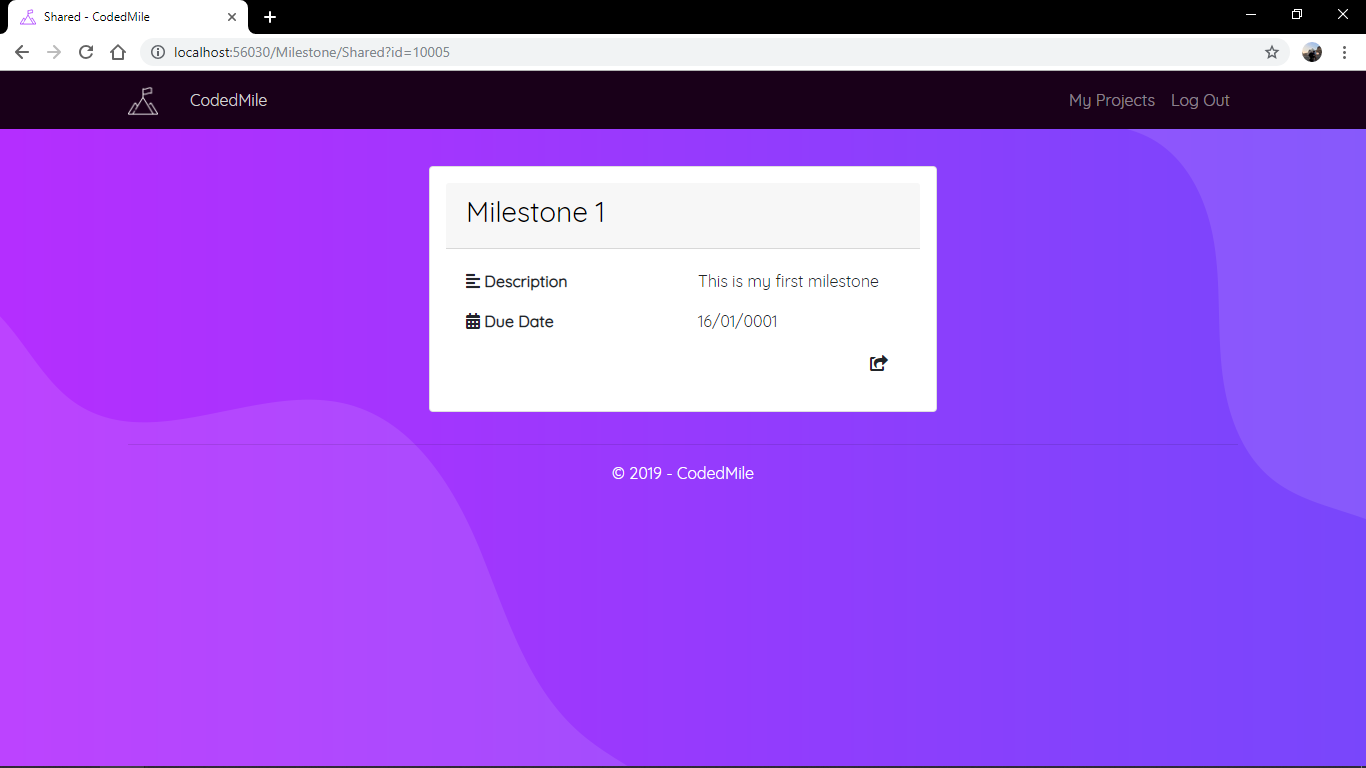
1.7)



1.8)

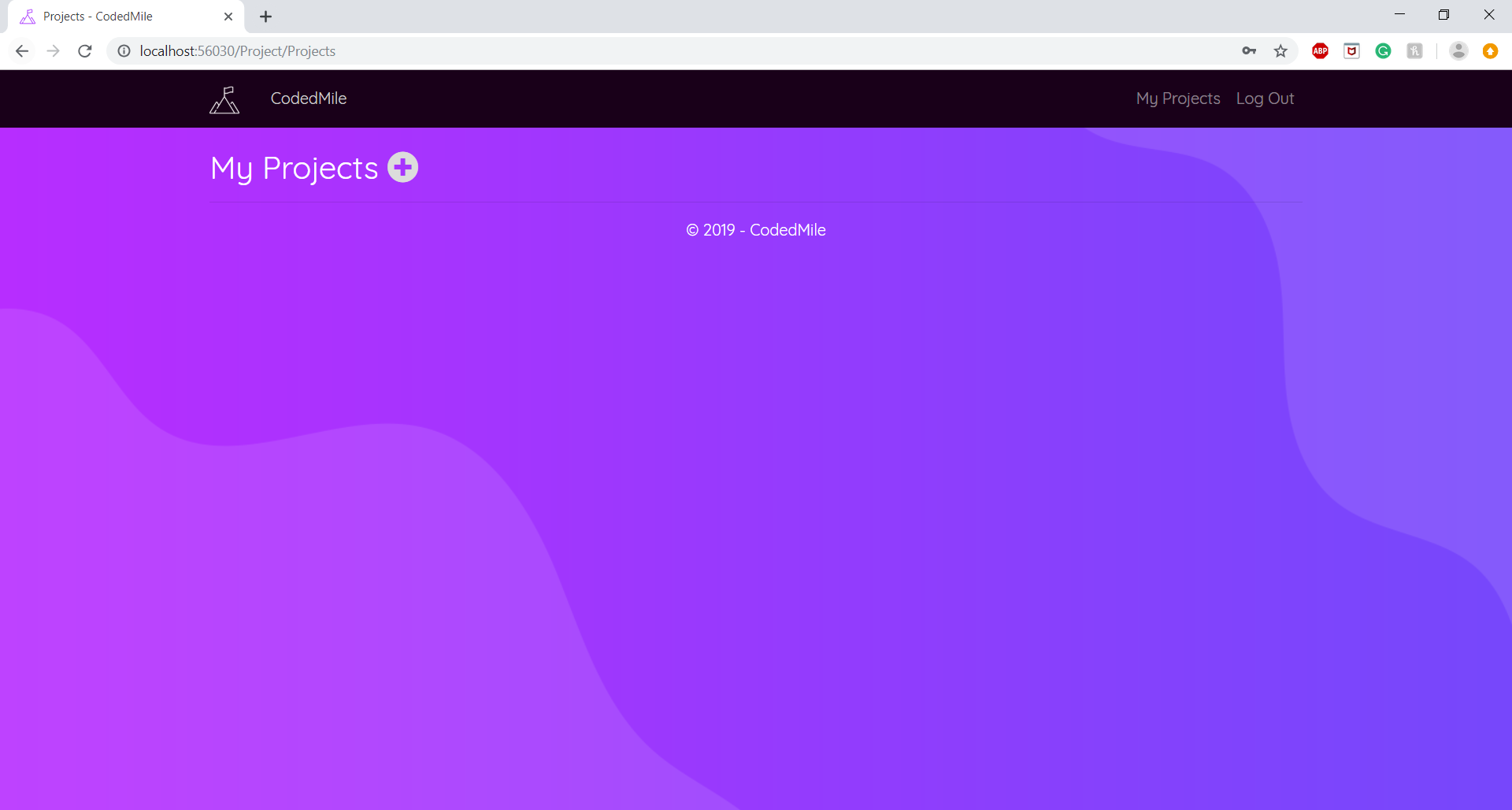
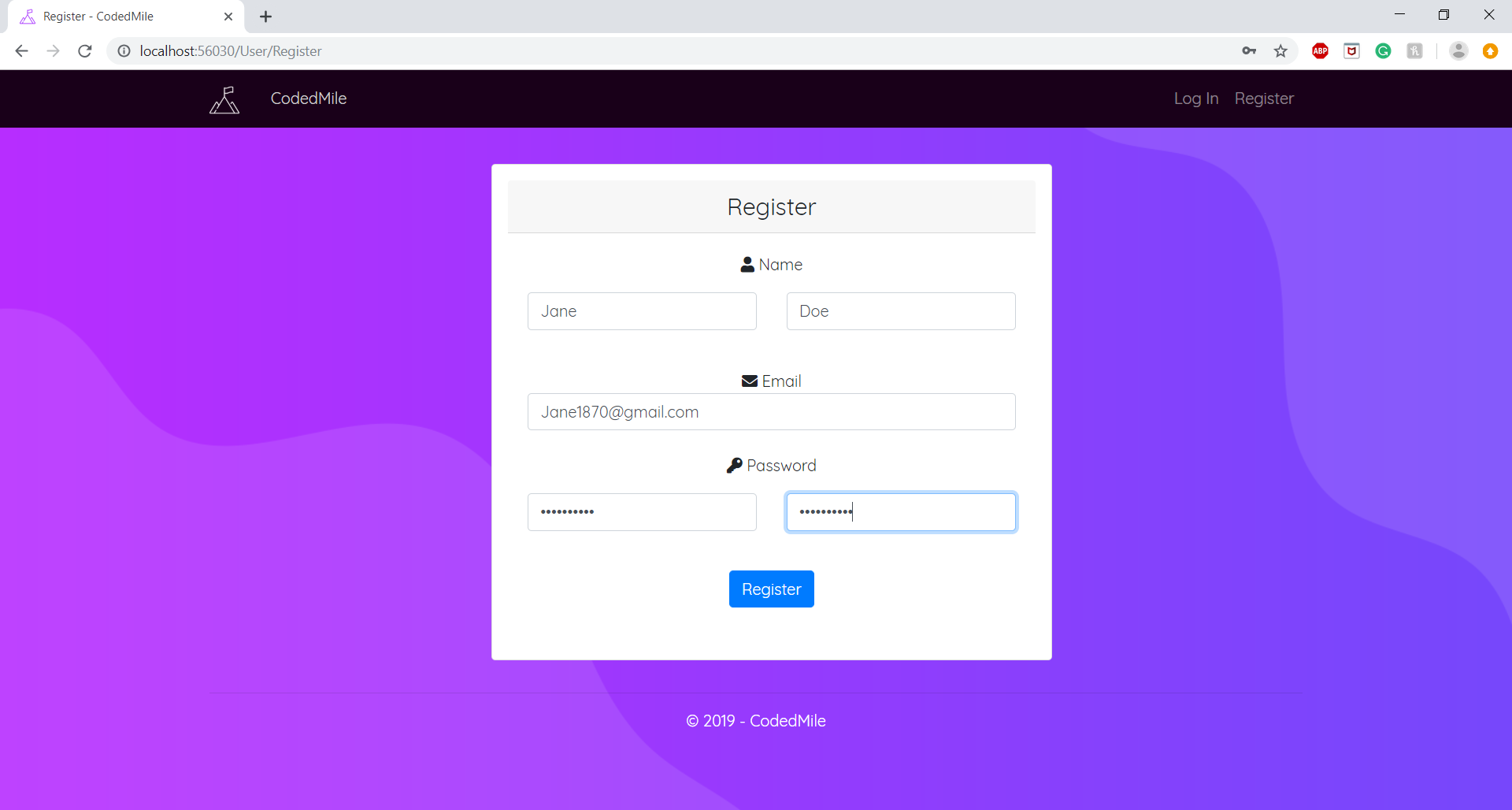


1.9)

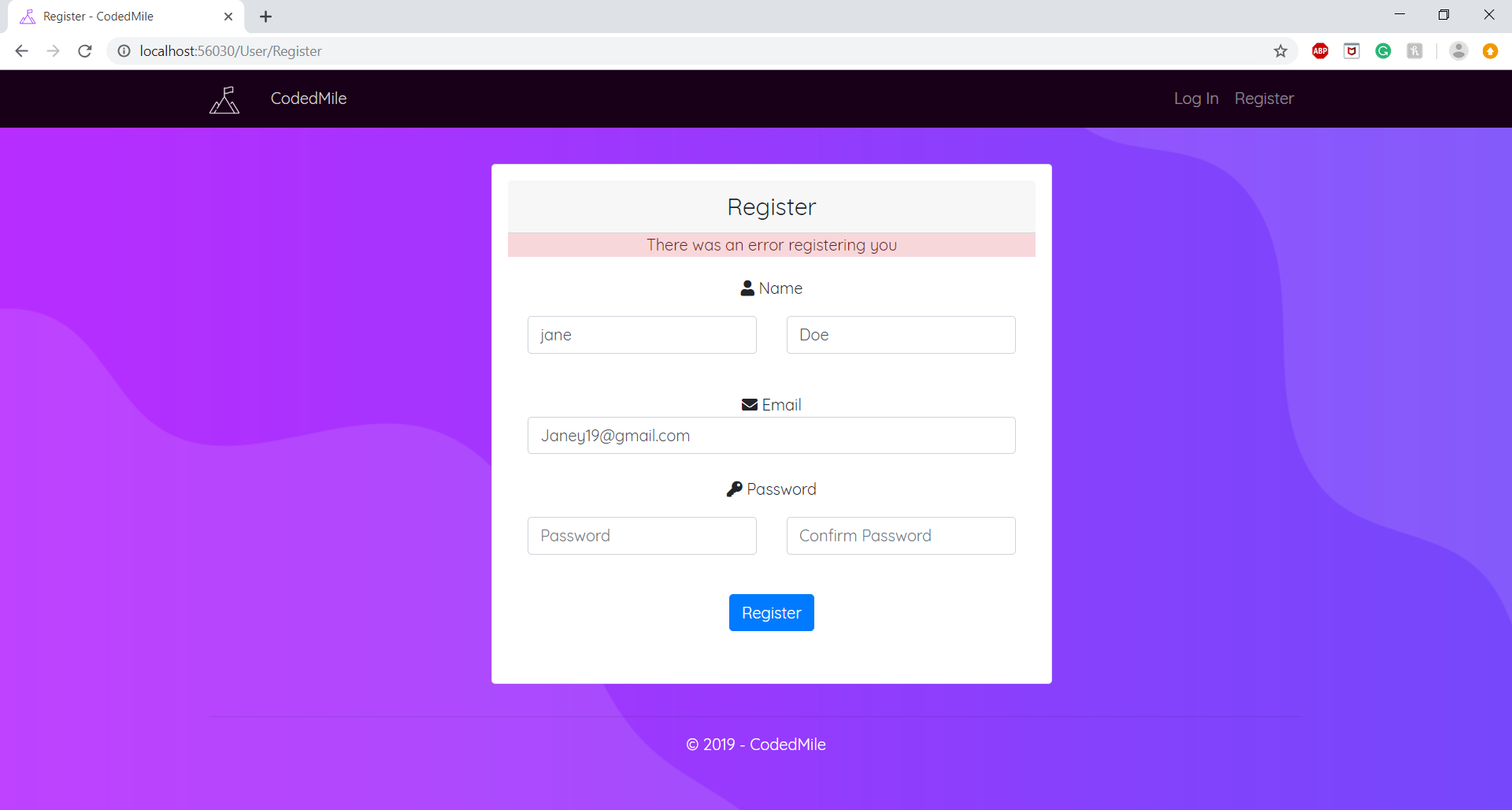


## 2 - Input Validation Testing

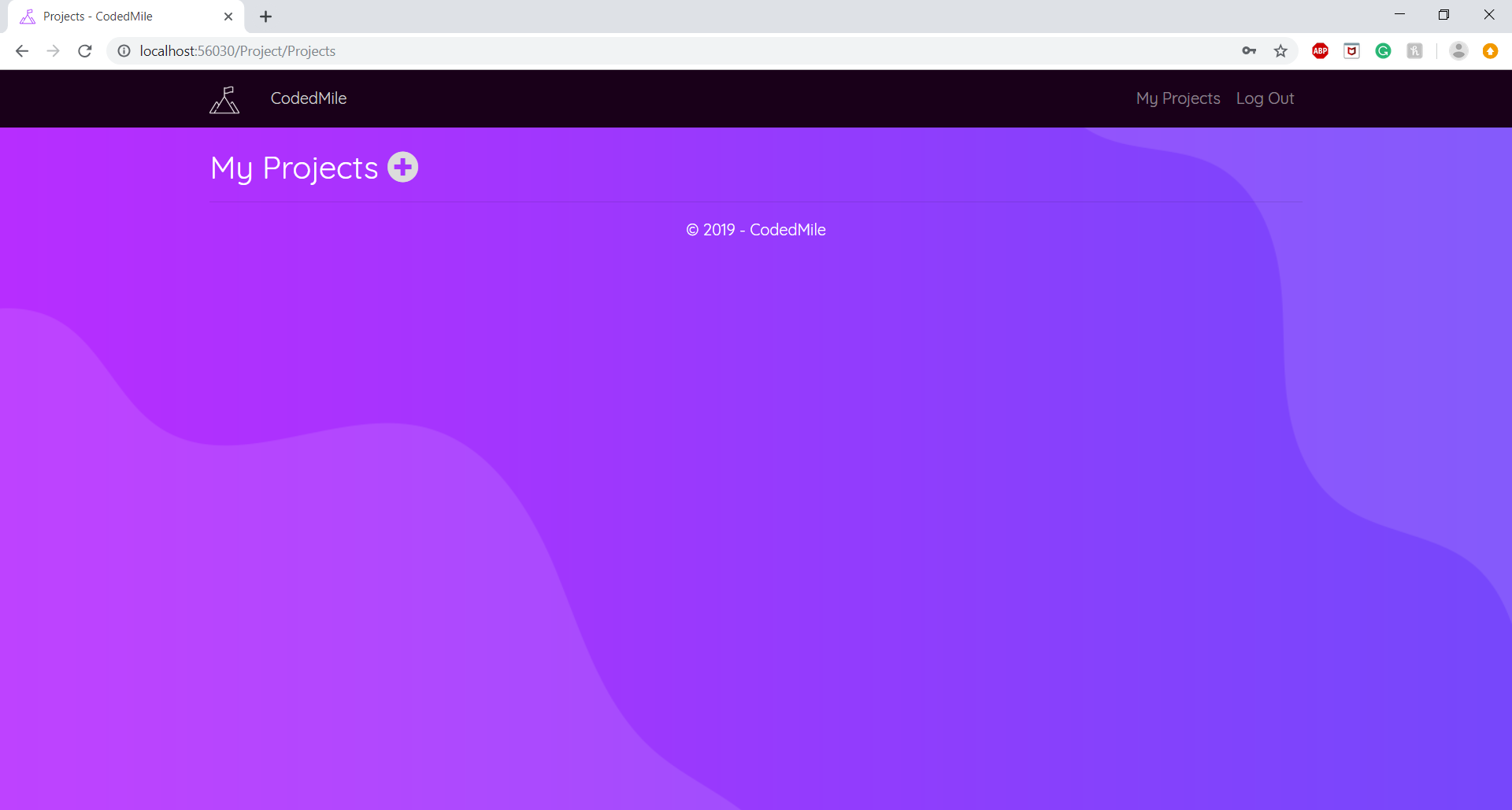
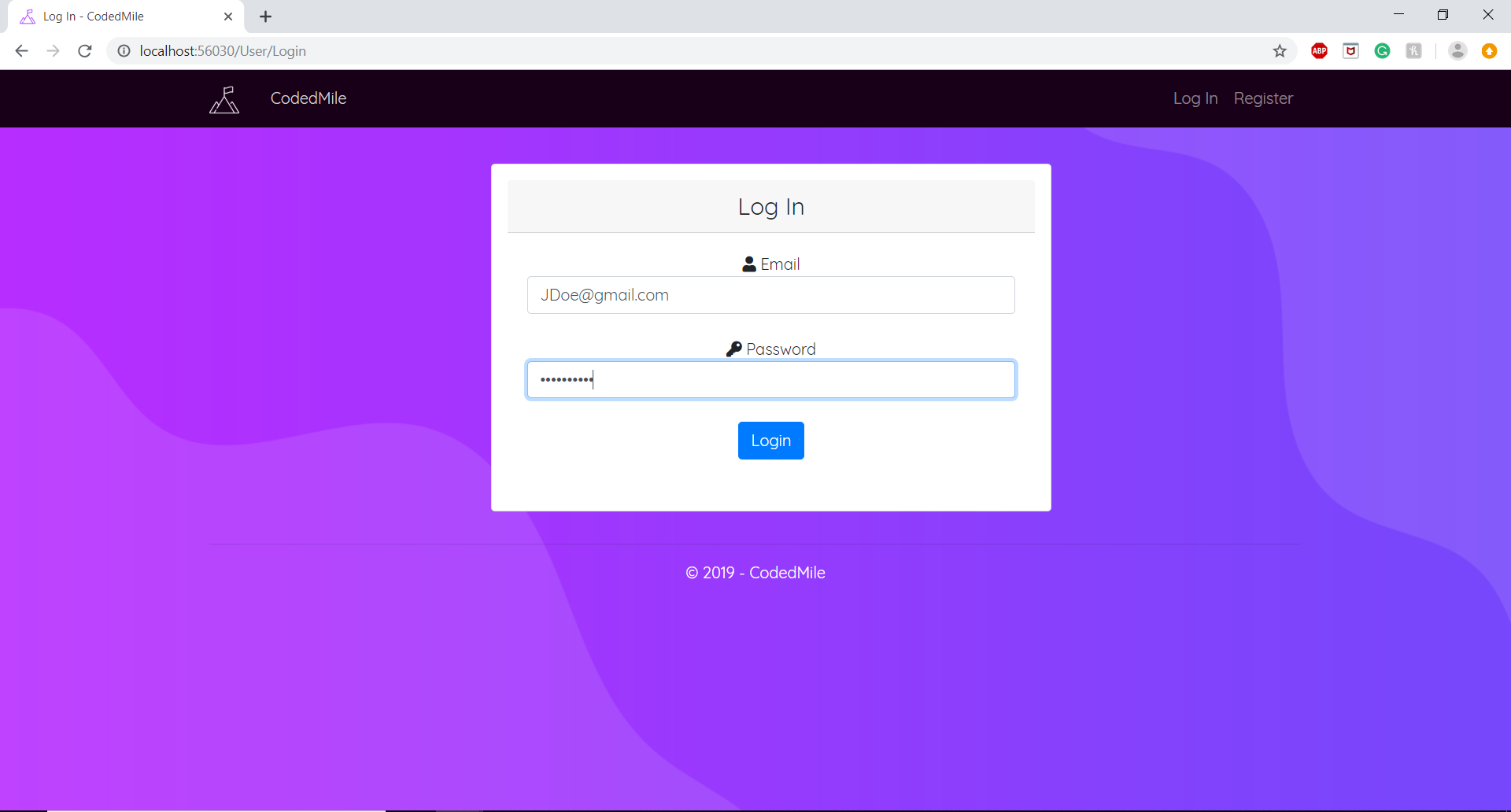
2.1)



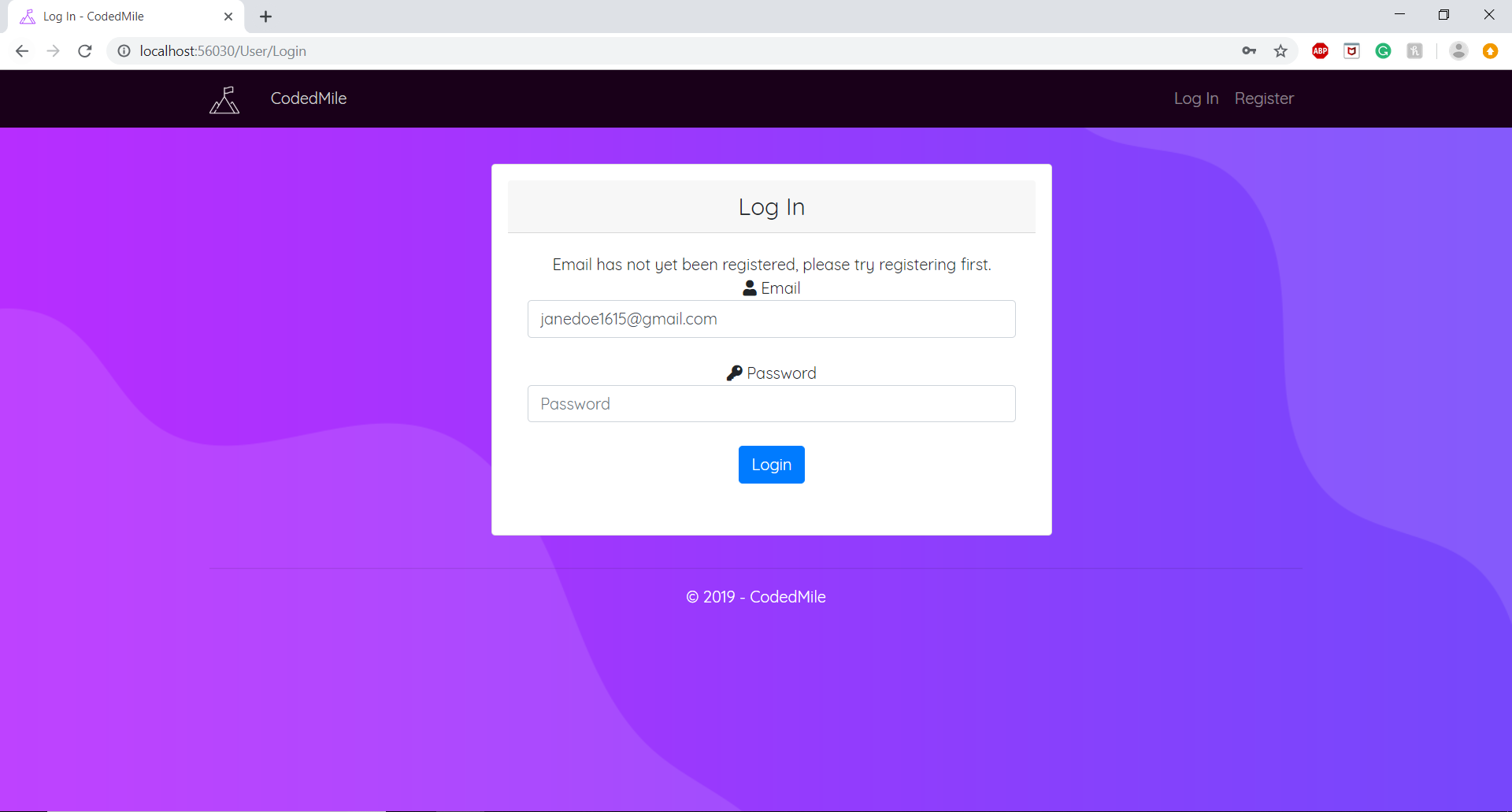
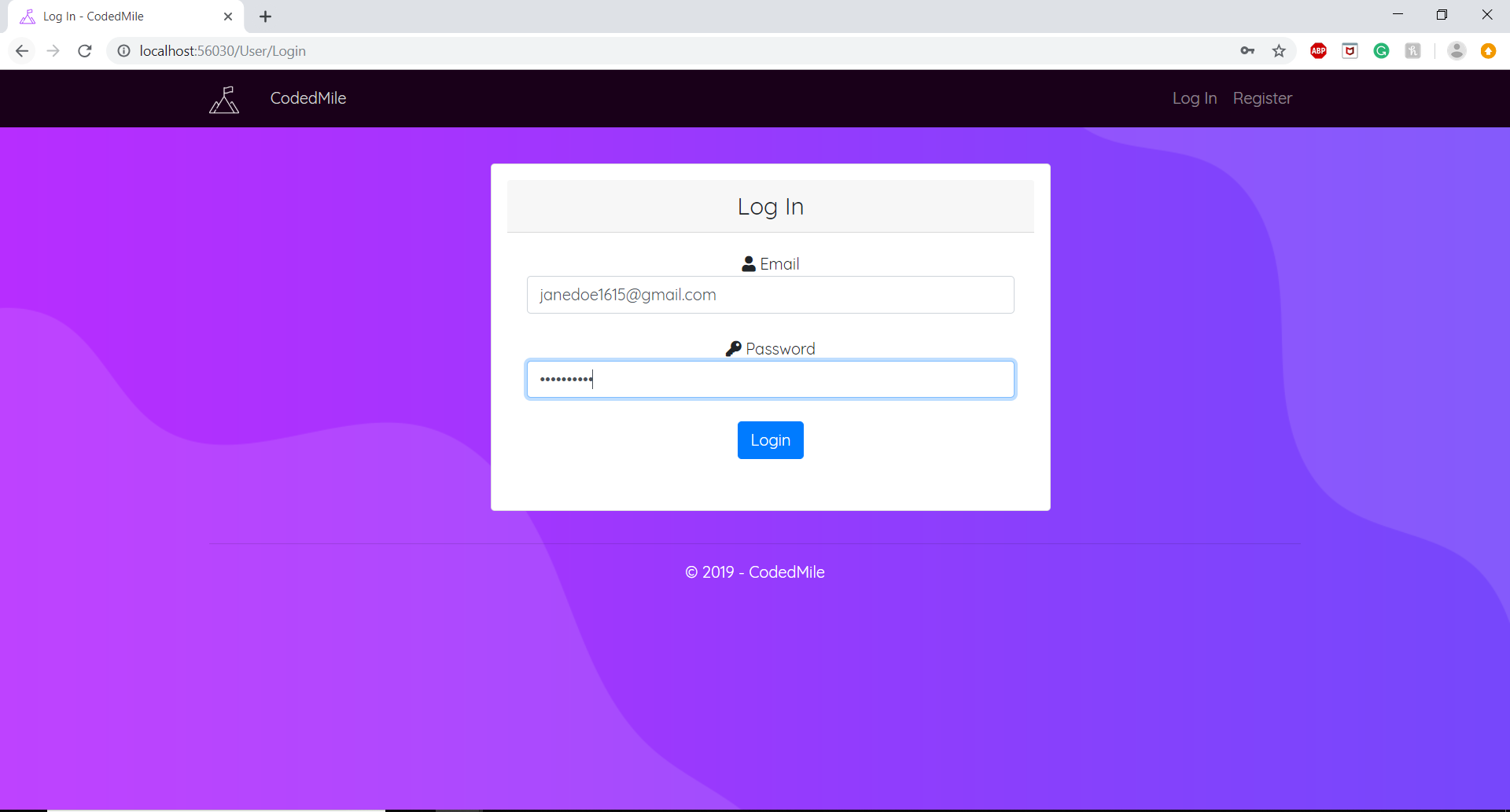
2.2)



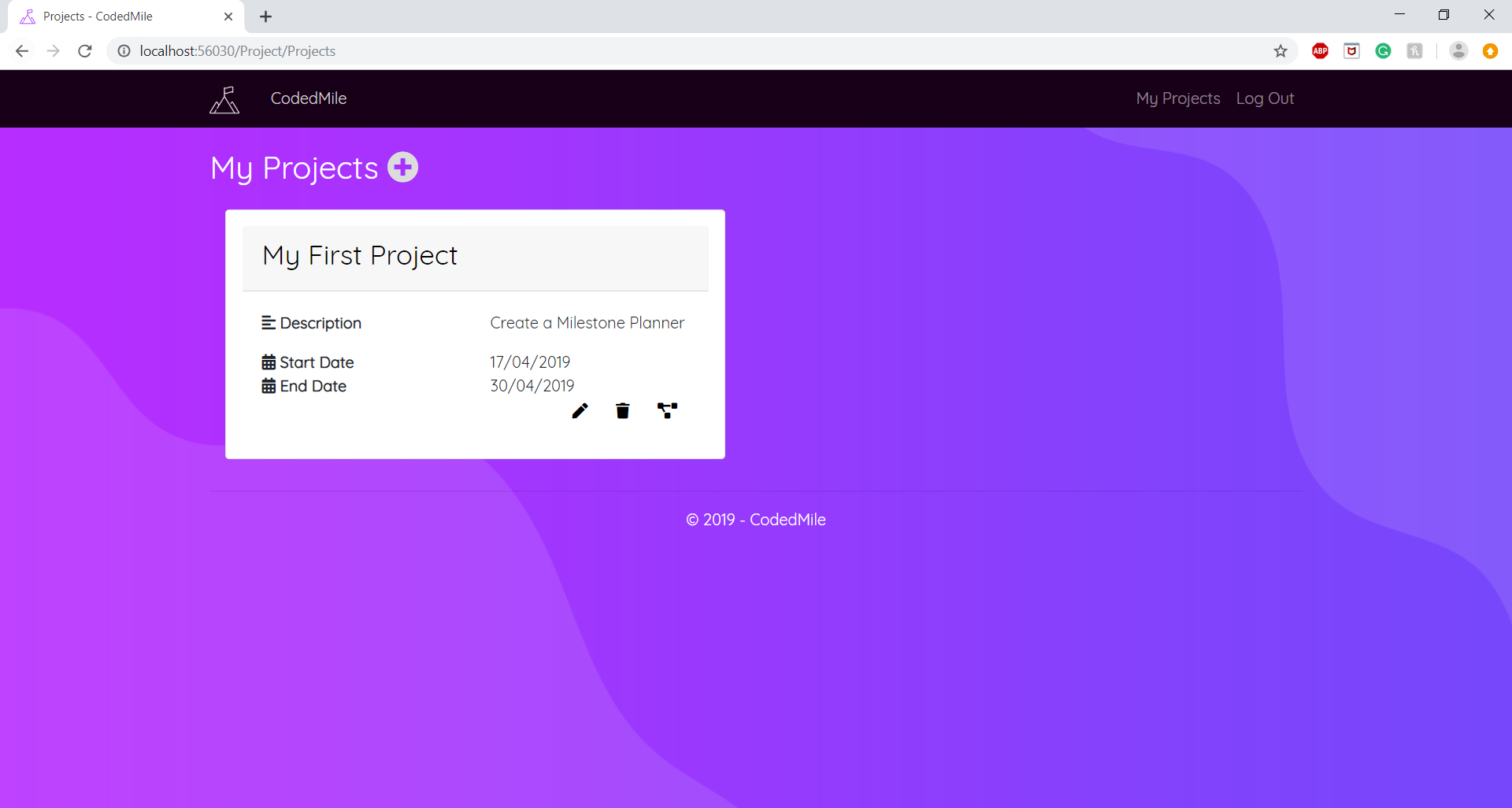
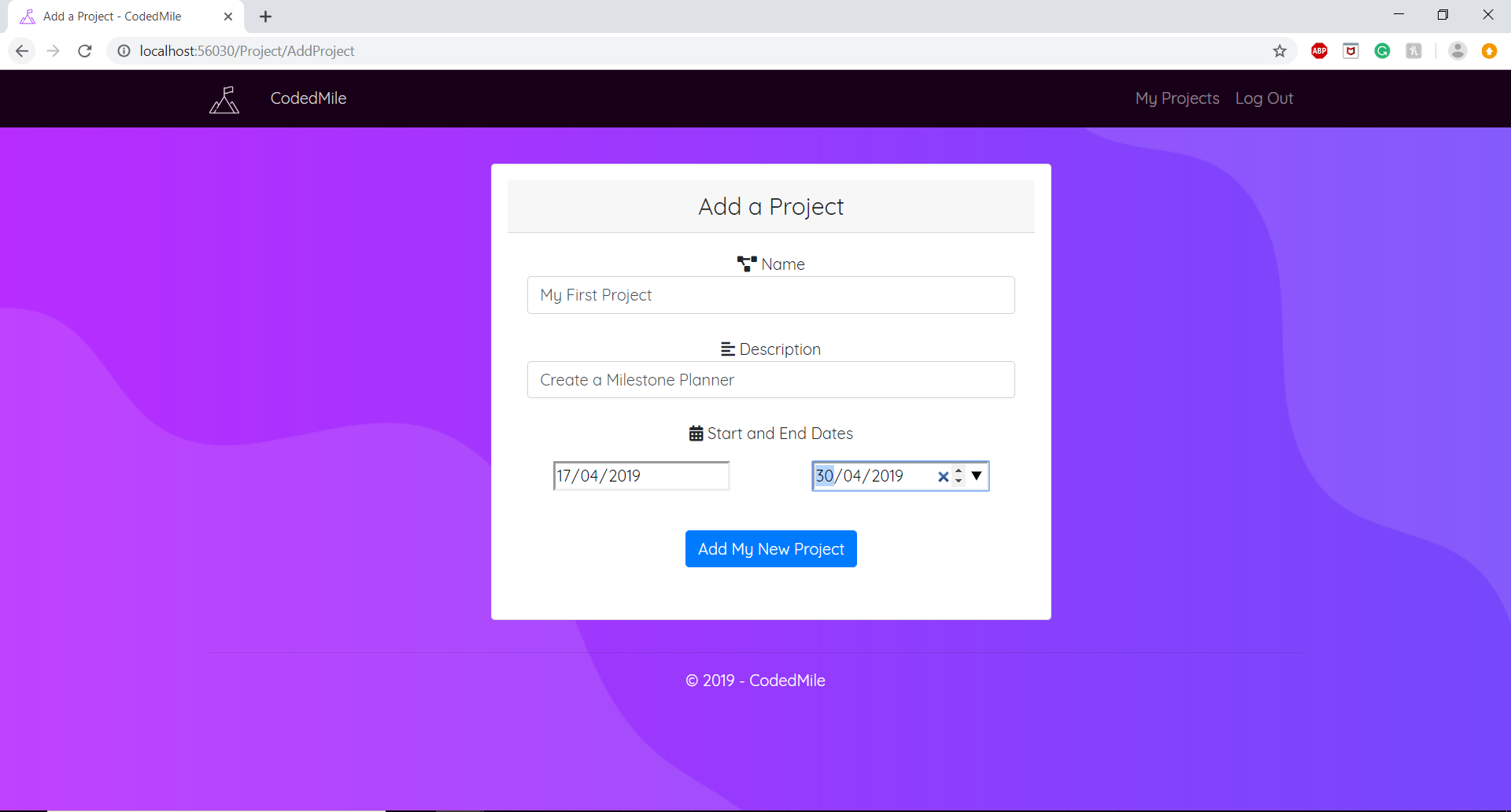
2.3)



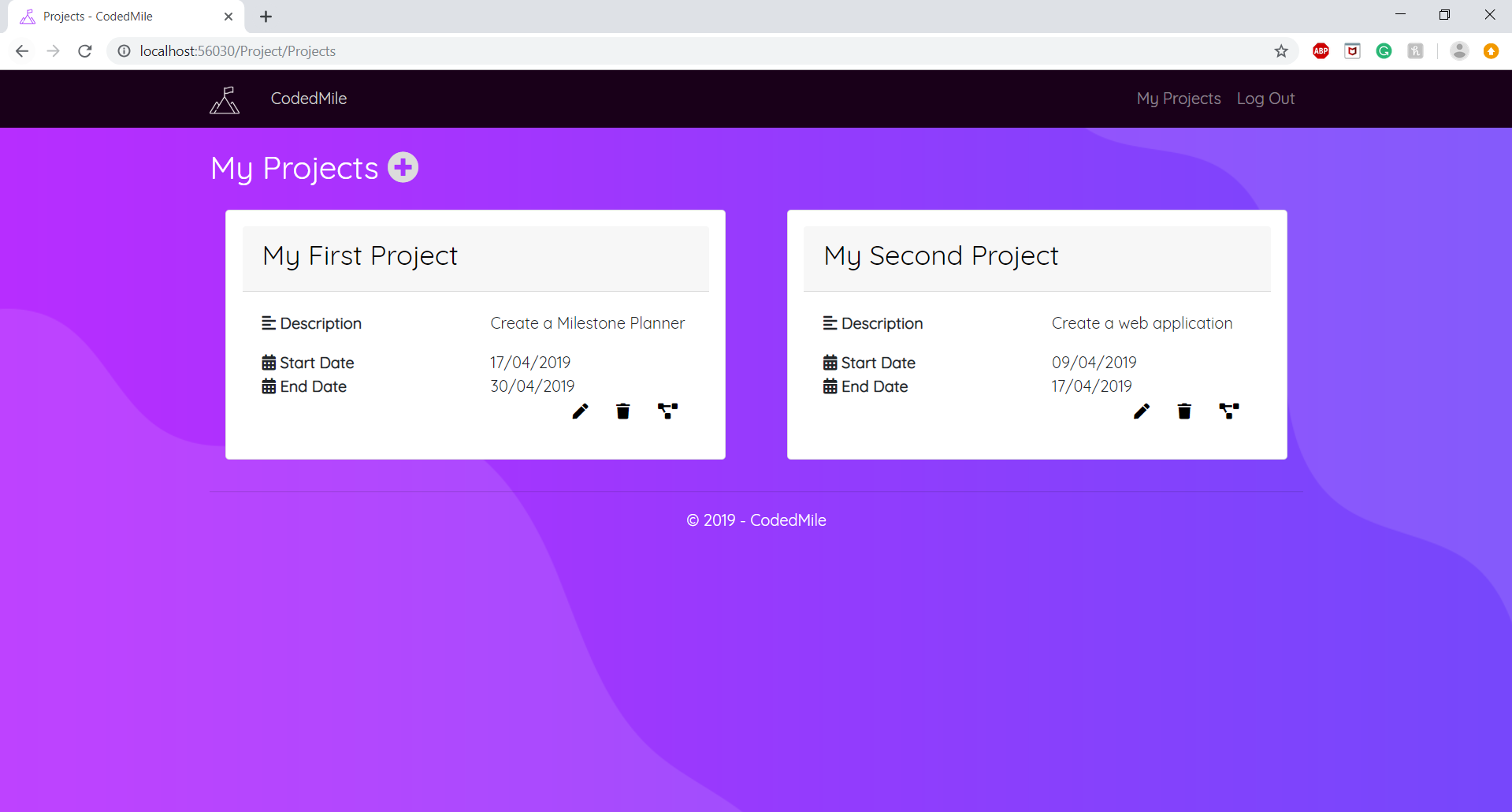
2.4)



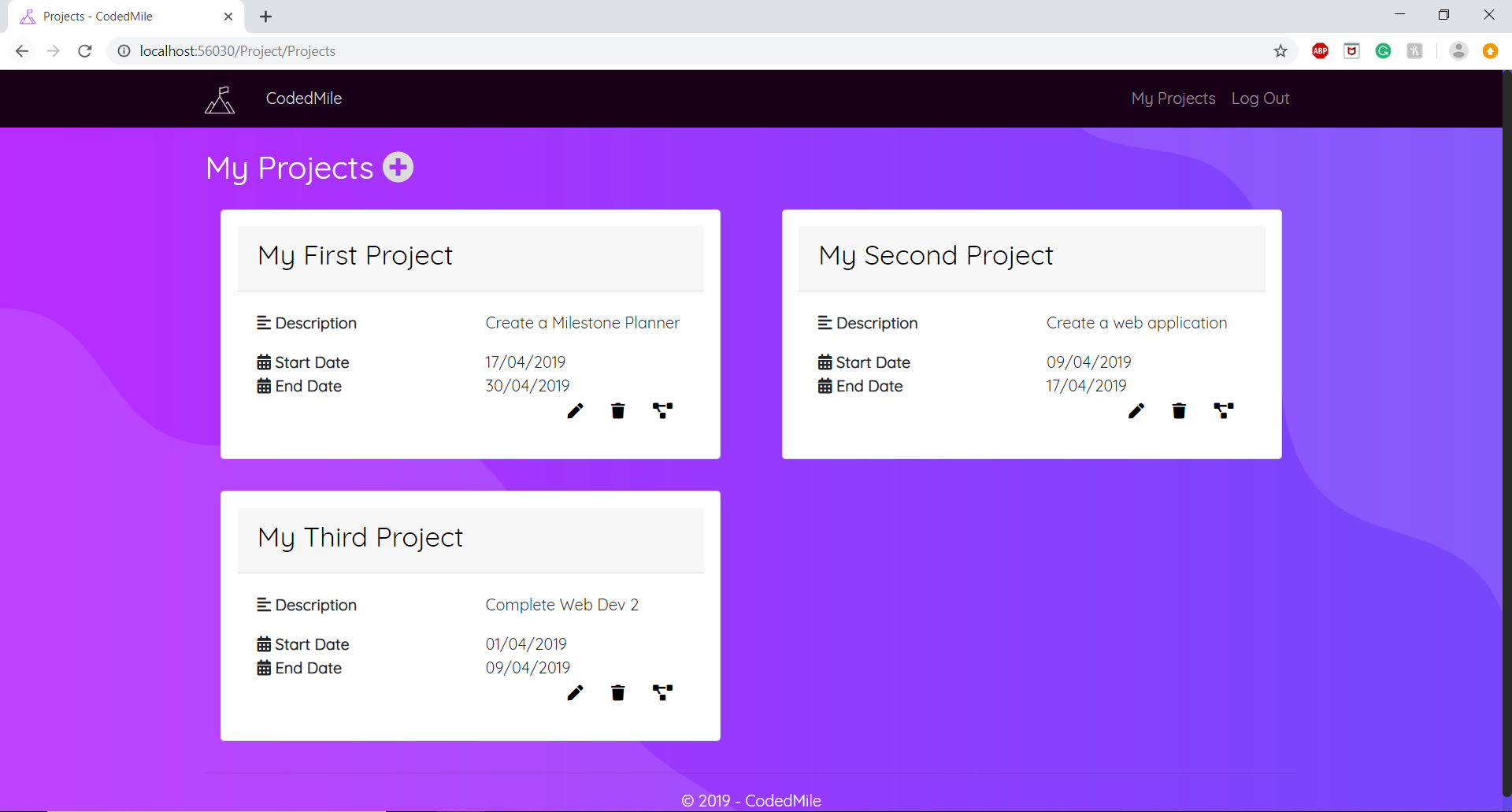
2.5)



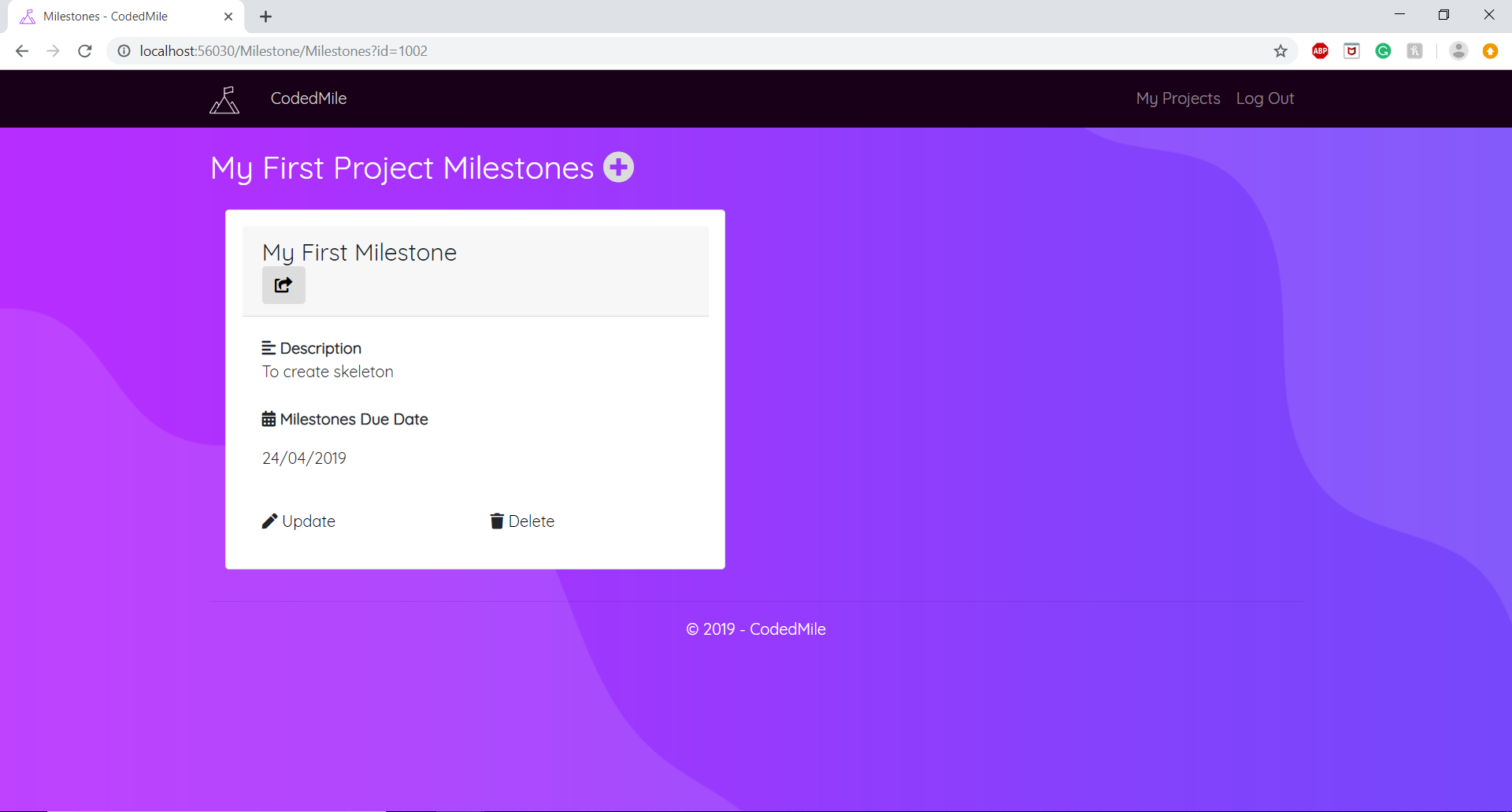
2.6)



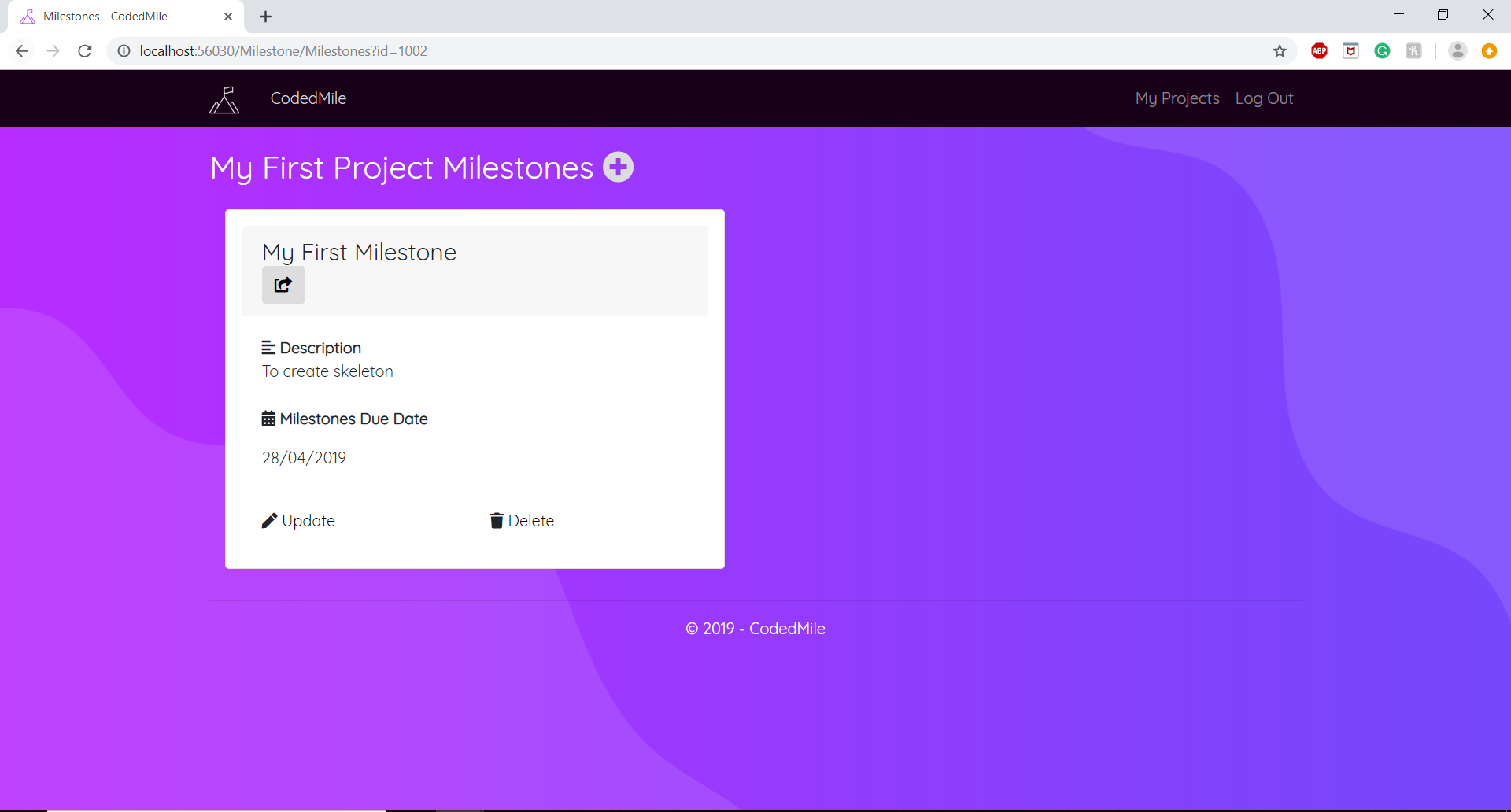
2.7)



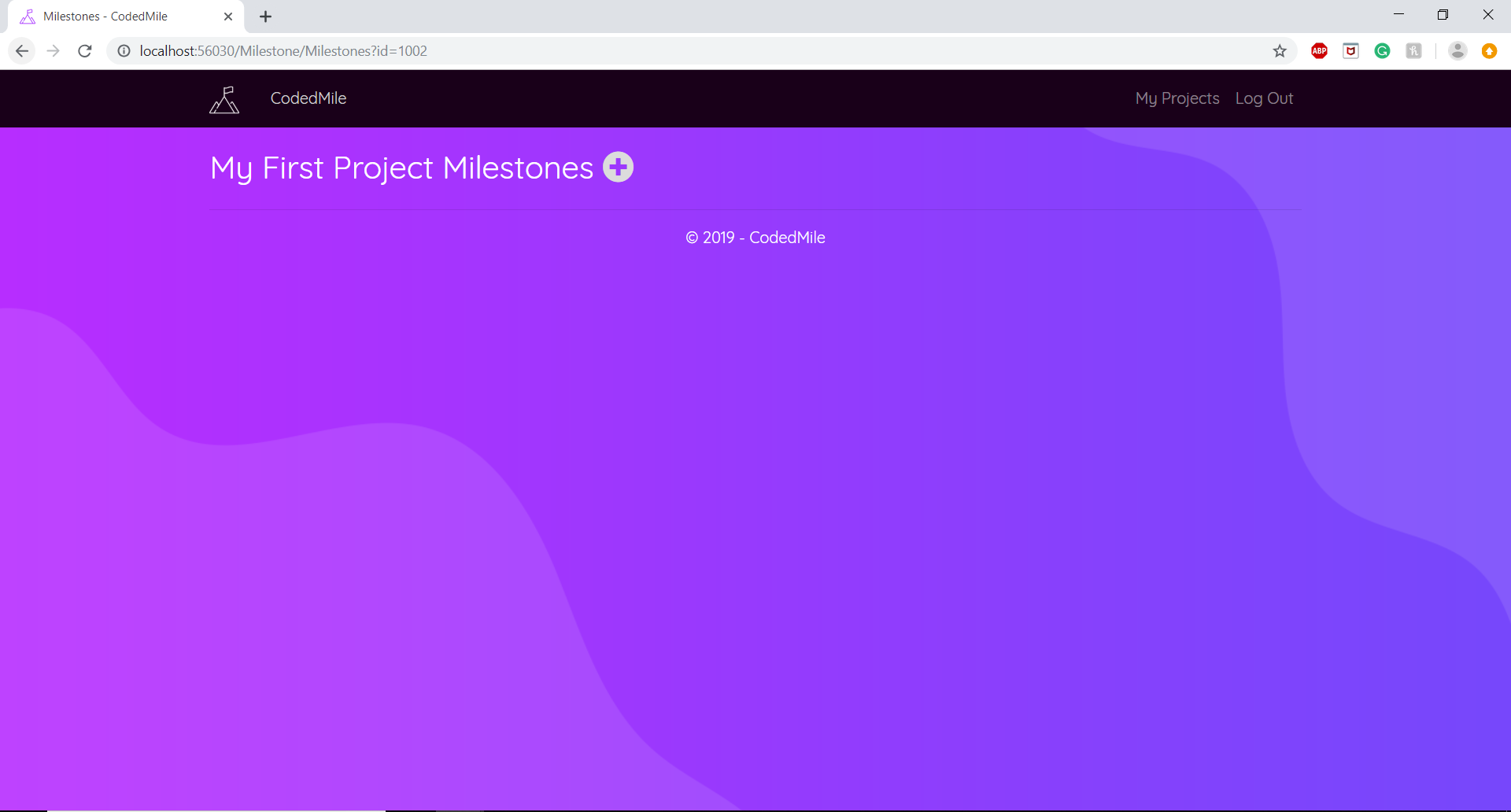
2.8)



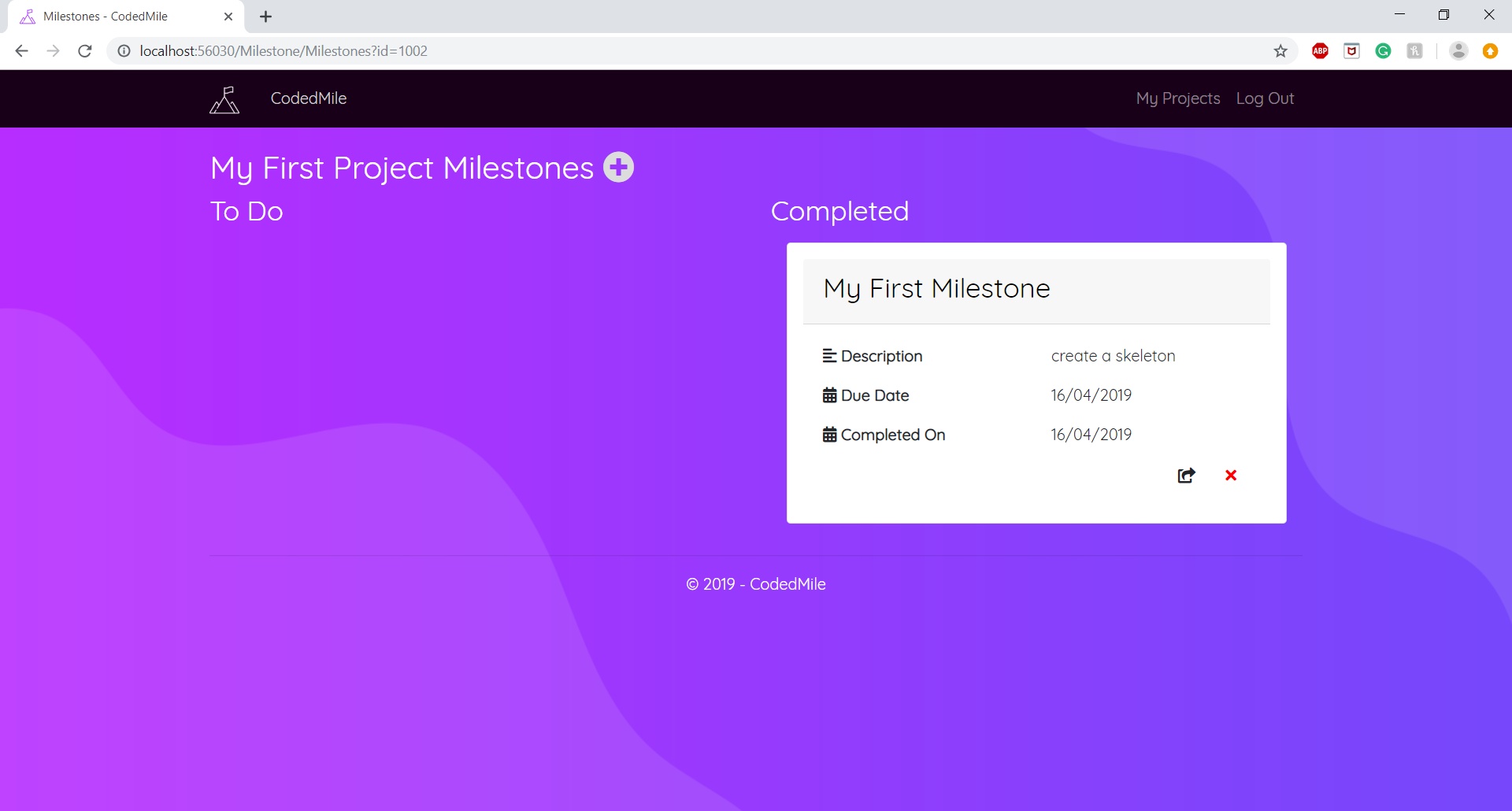
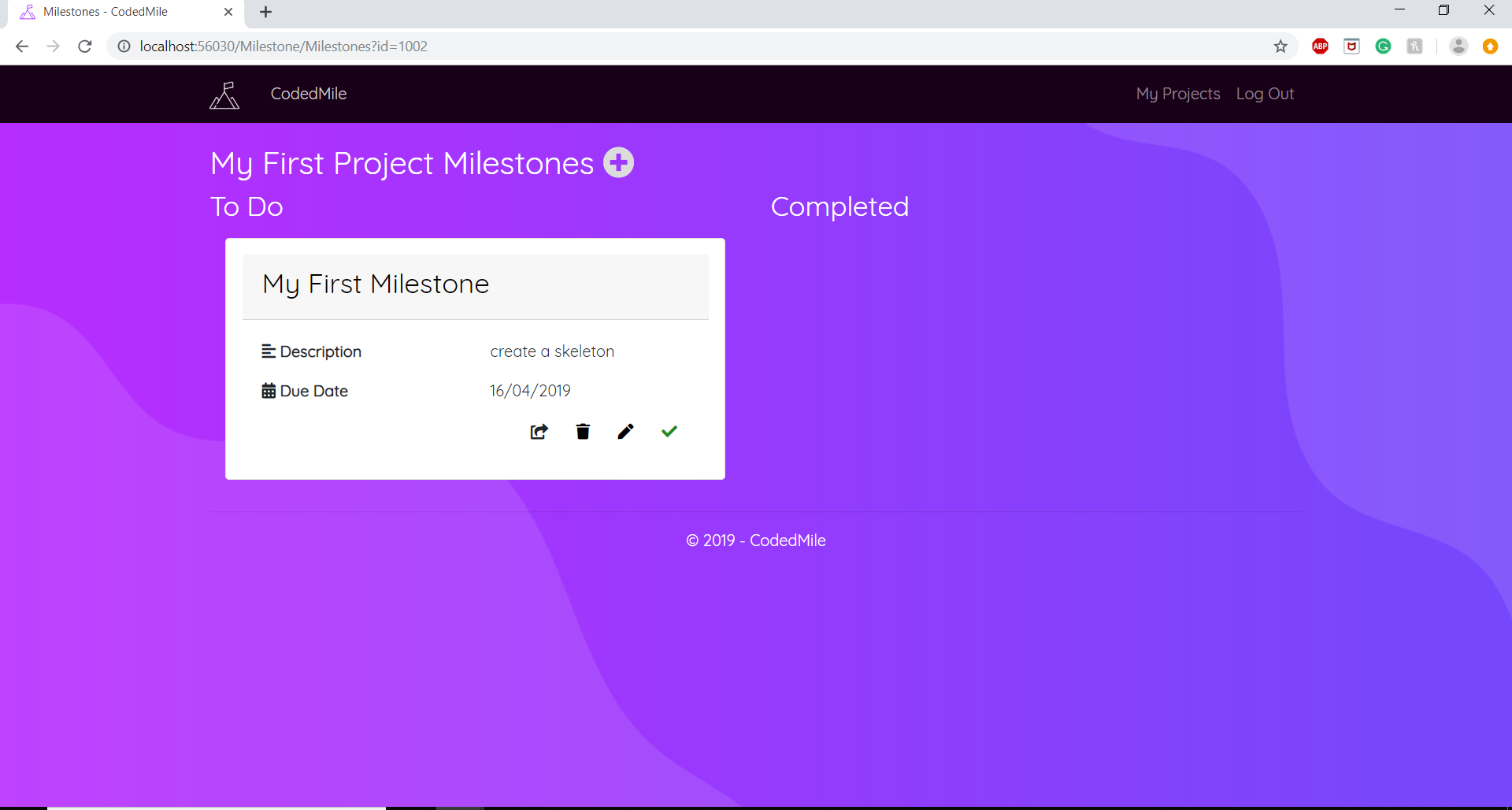
2.9)



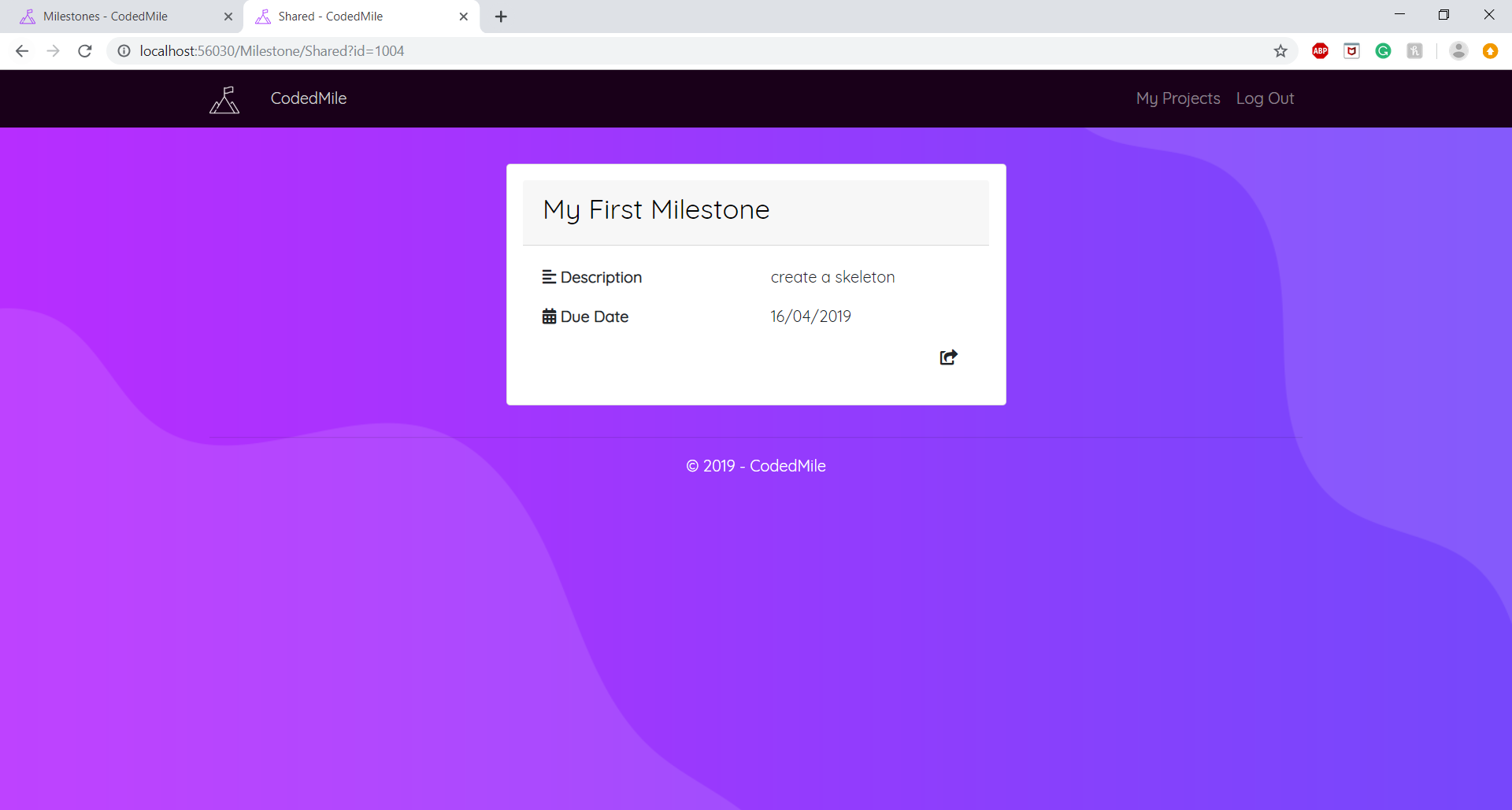
2.10)



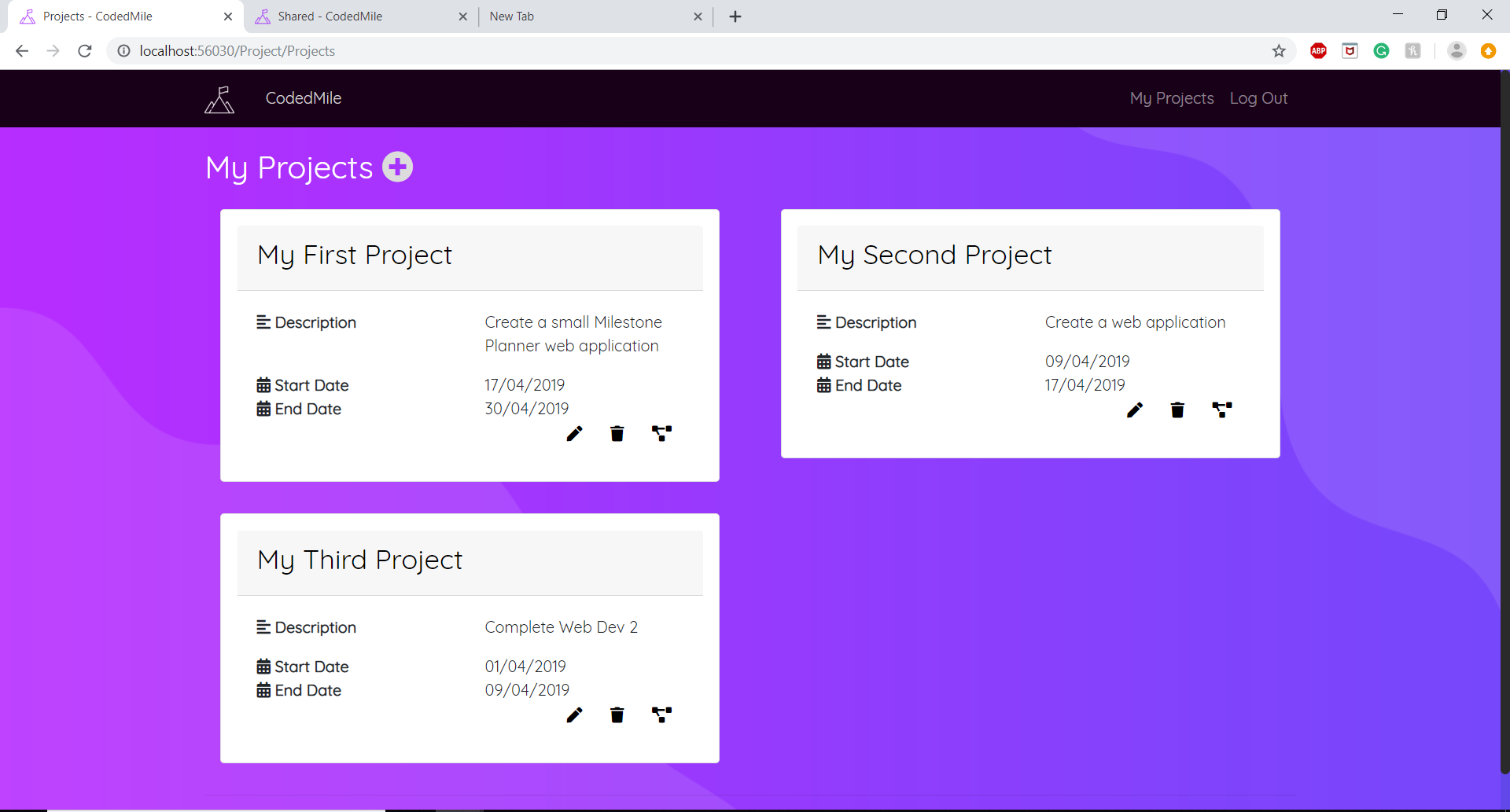
2.11)



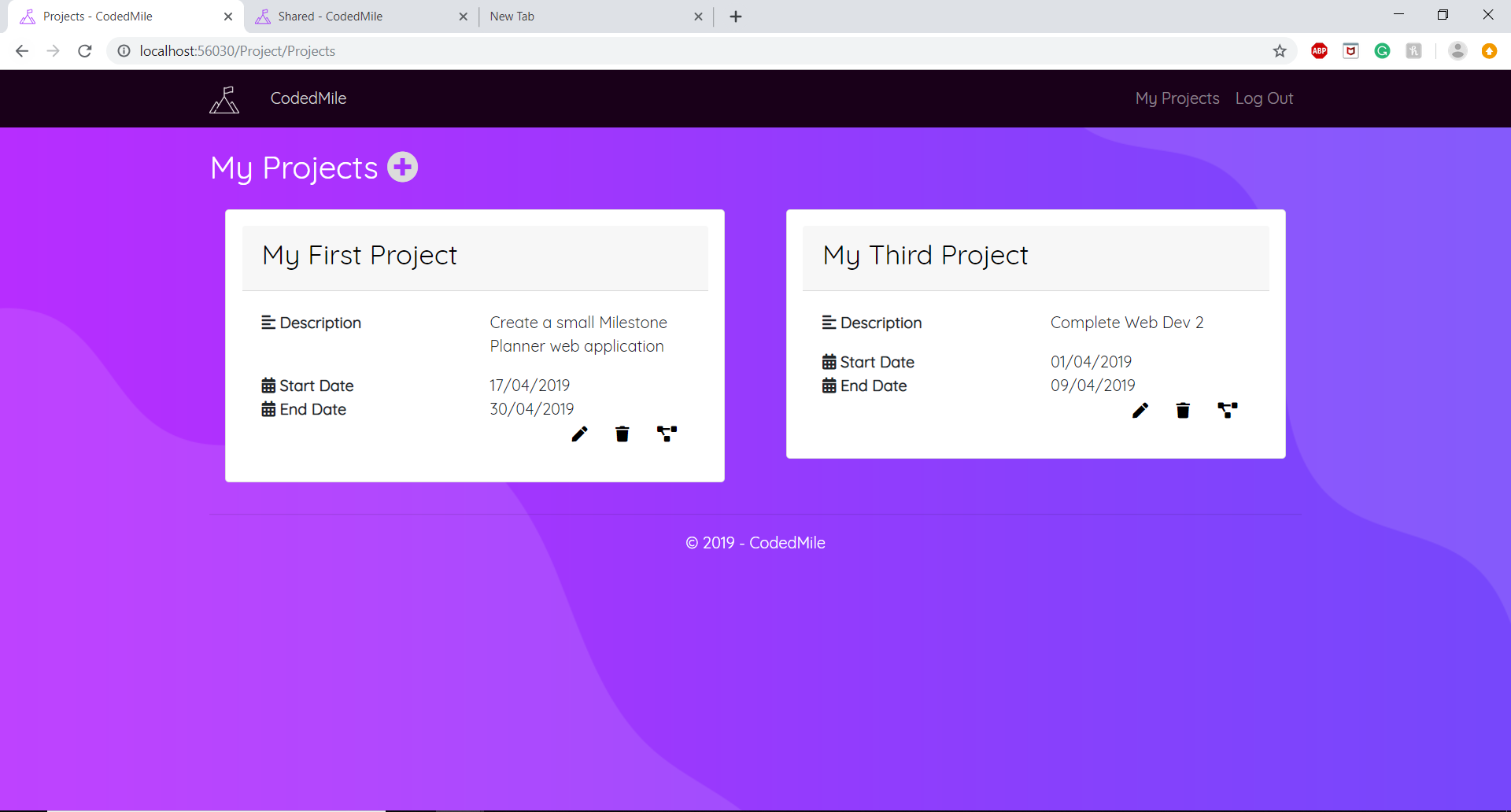
2.12)



2.13)

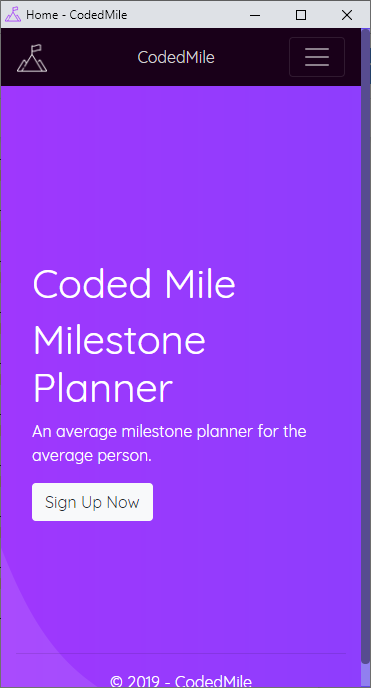
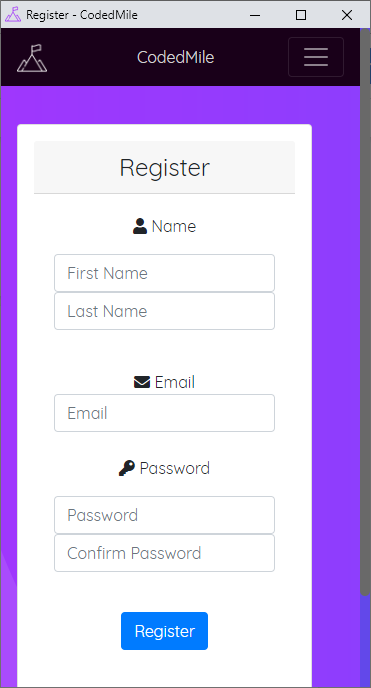
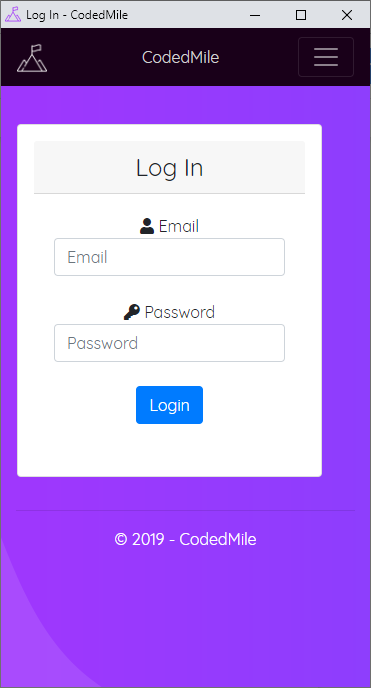
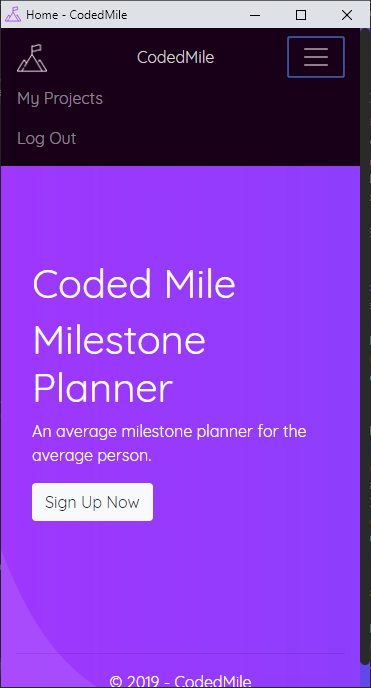


2.14)

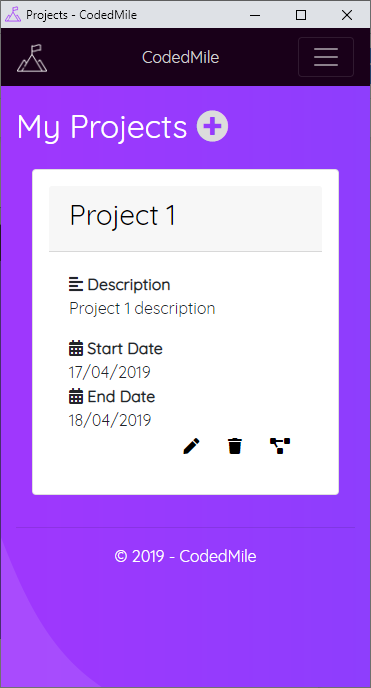
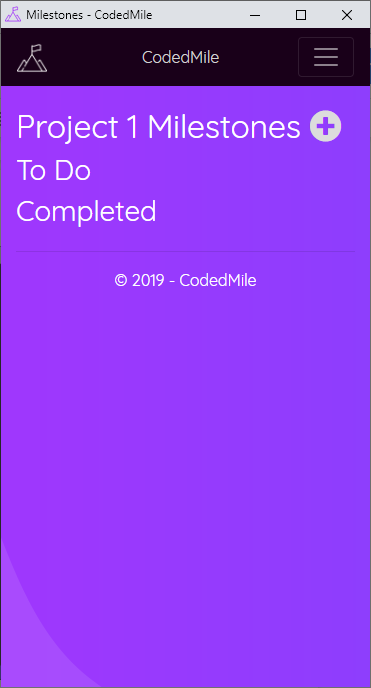
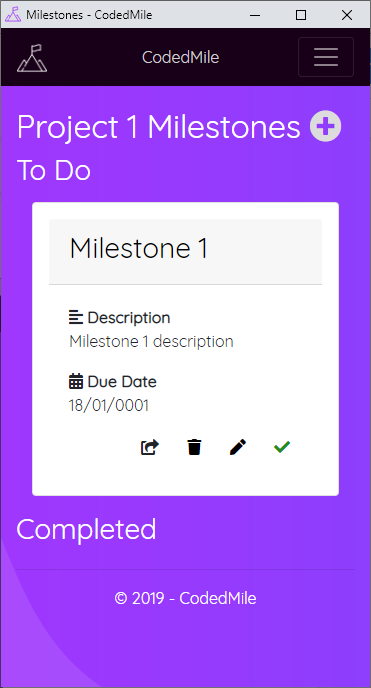


## 3 - Responsiveness Testing

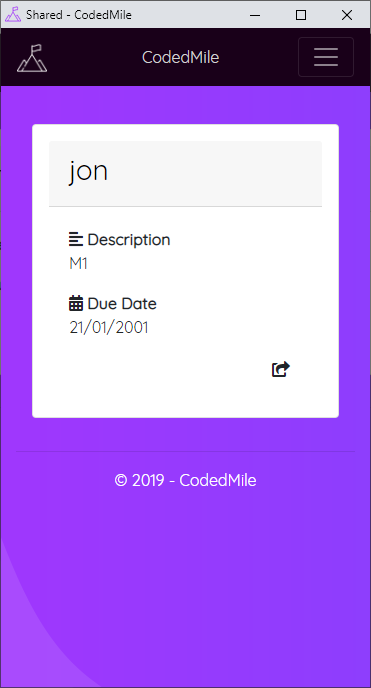
3.1) 3.2) 3.3) 3.4)

3.5) 3.6) 3.7) 3.8)

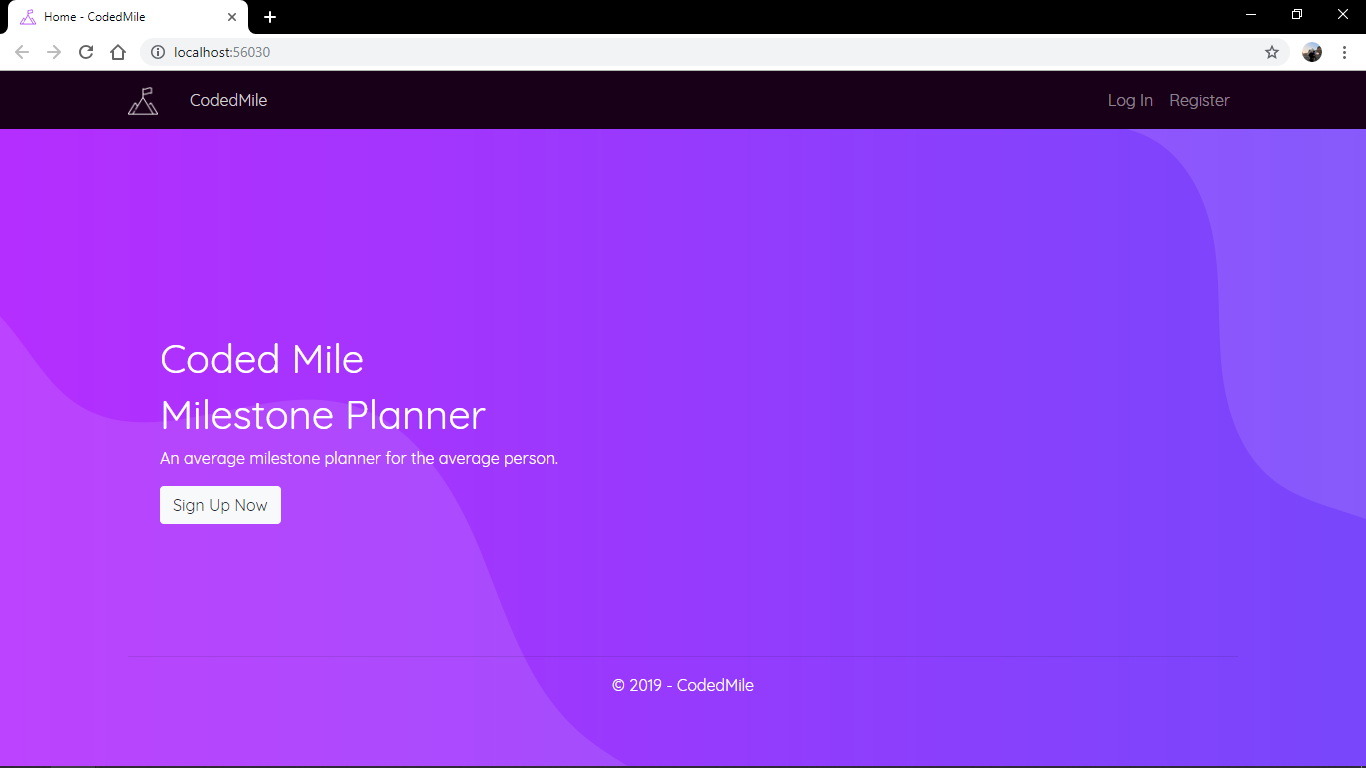
   

3.9)

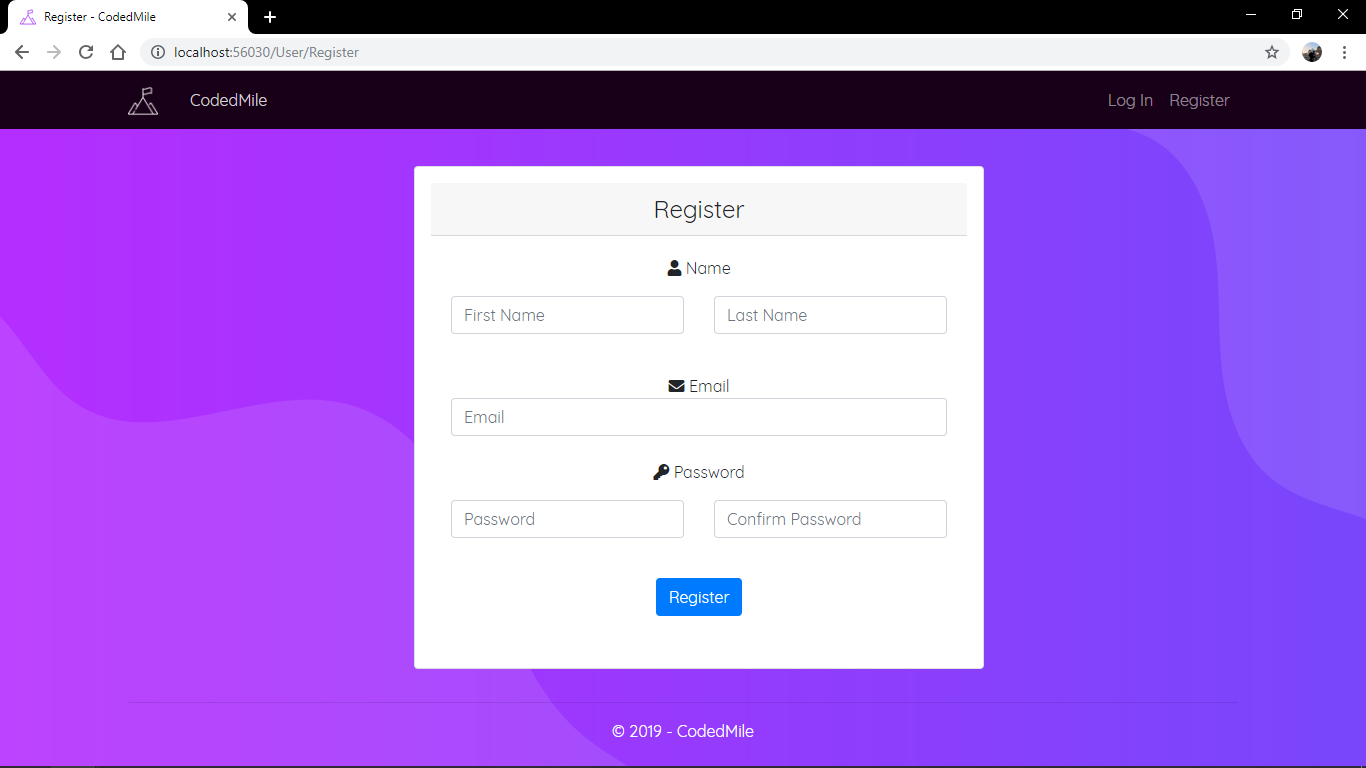


## 4 - User Access Testing

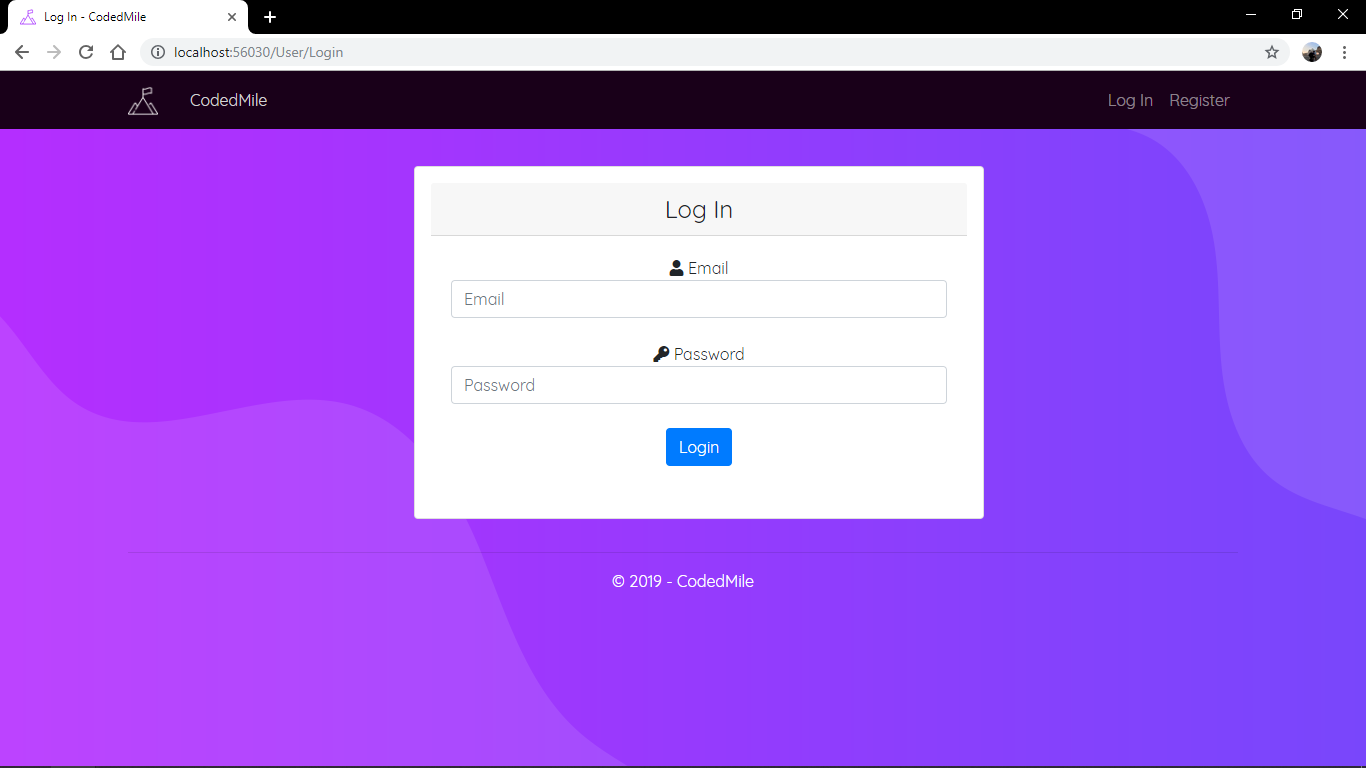
4.1)



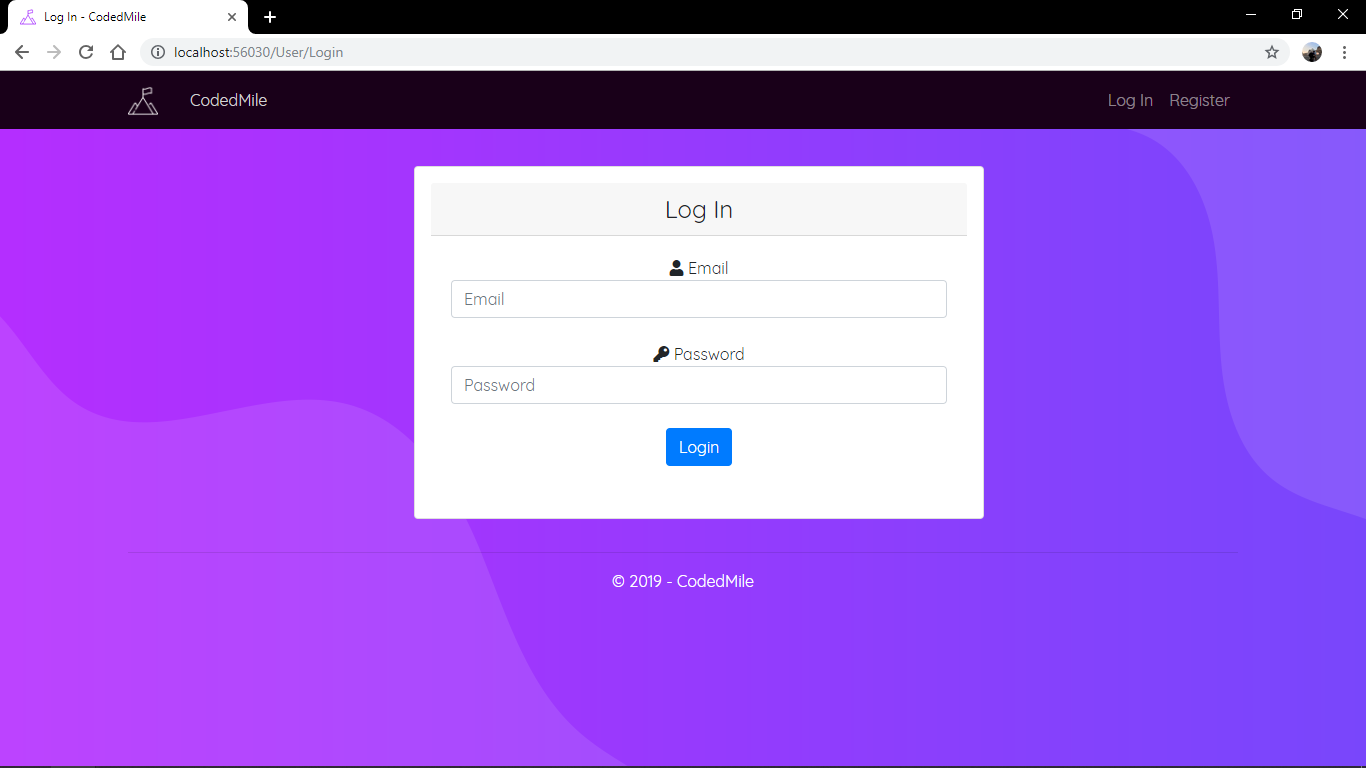
4.2)



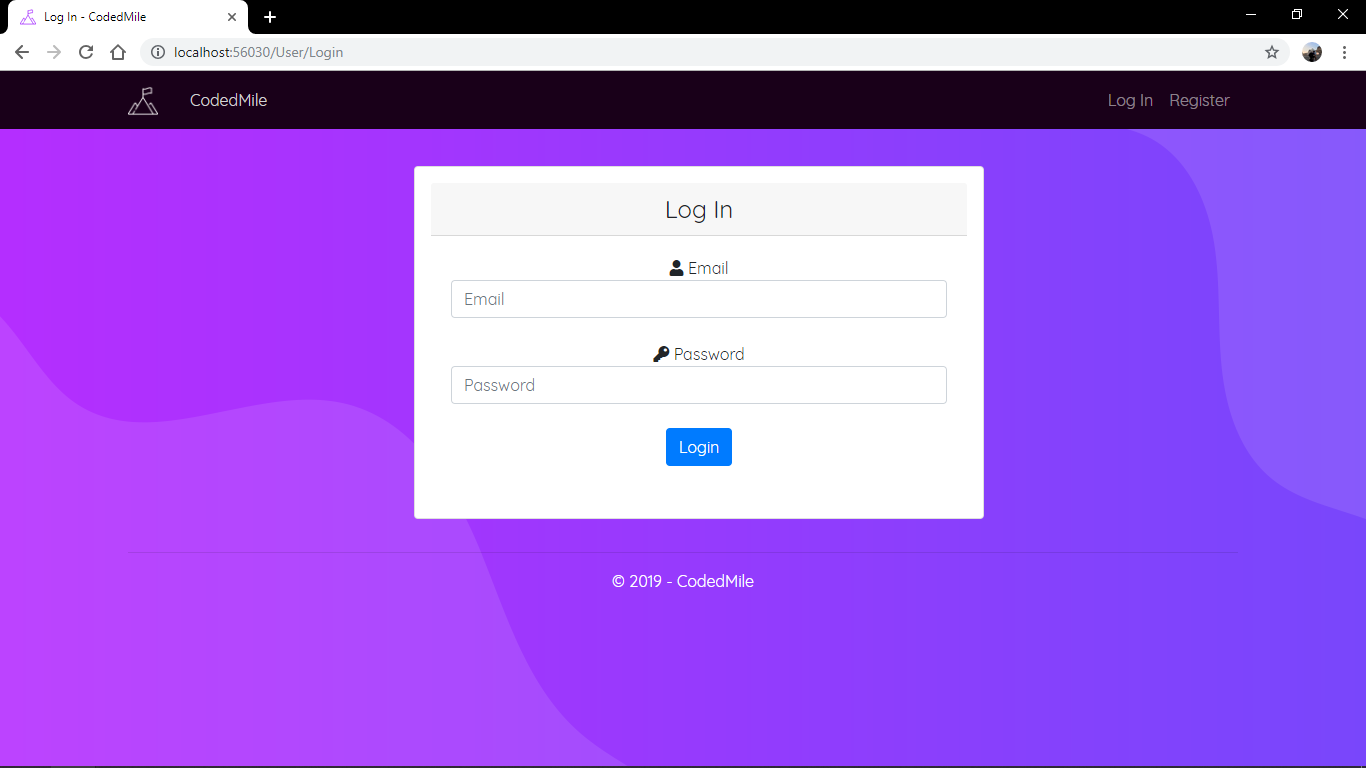
4.3)



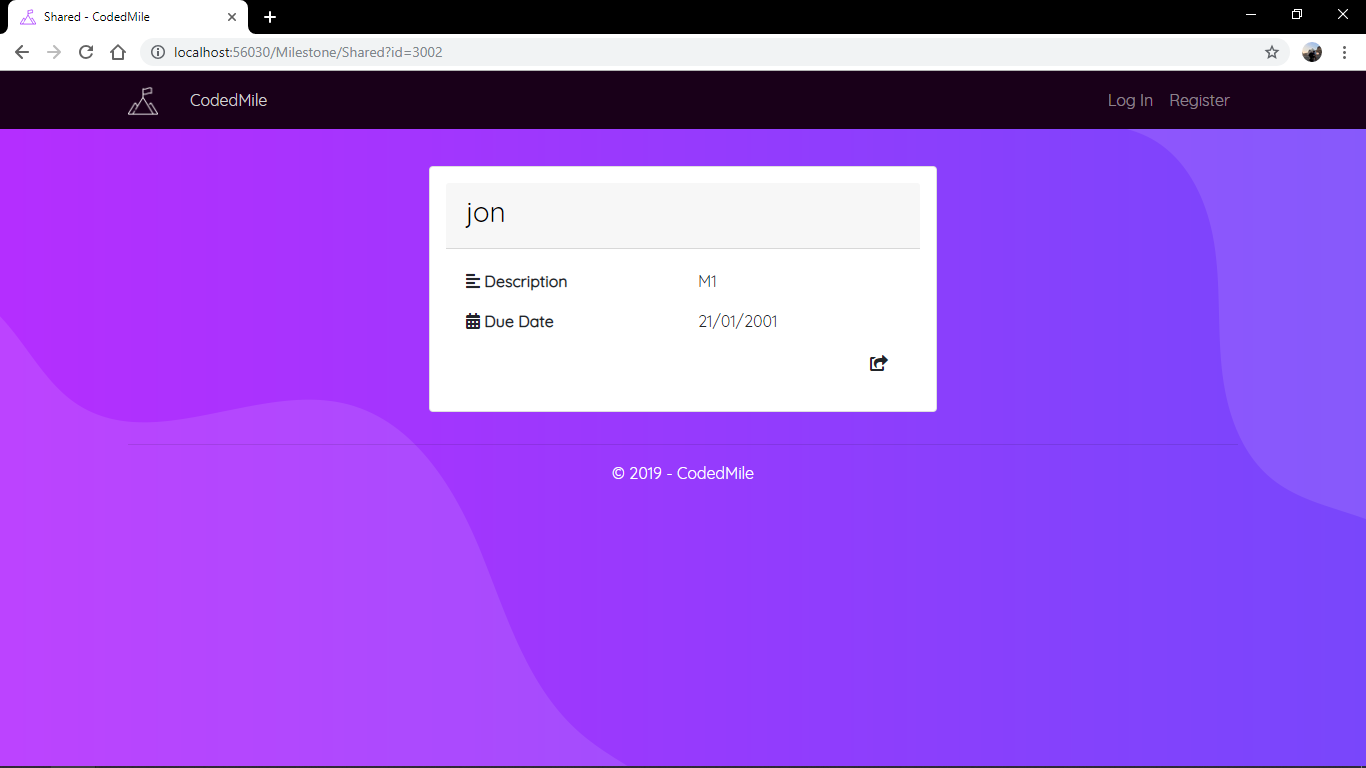
4.4)



4.5)

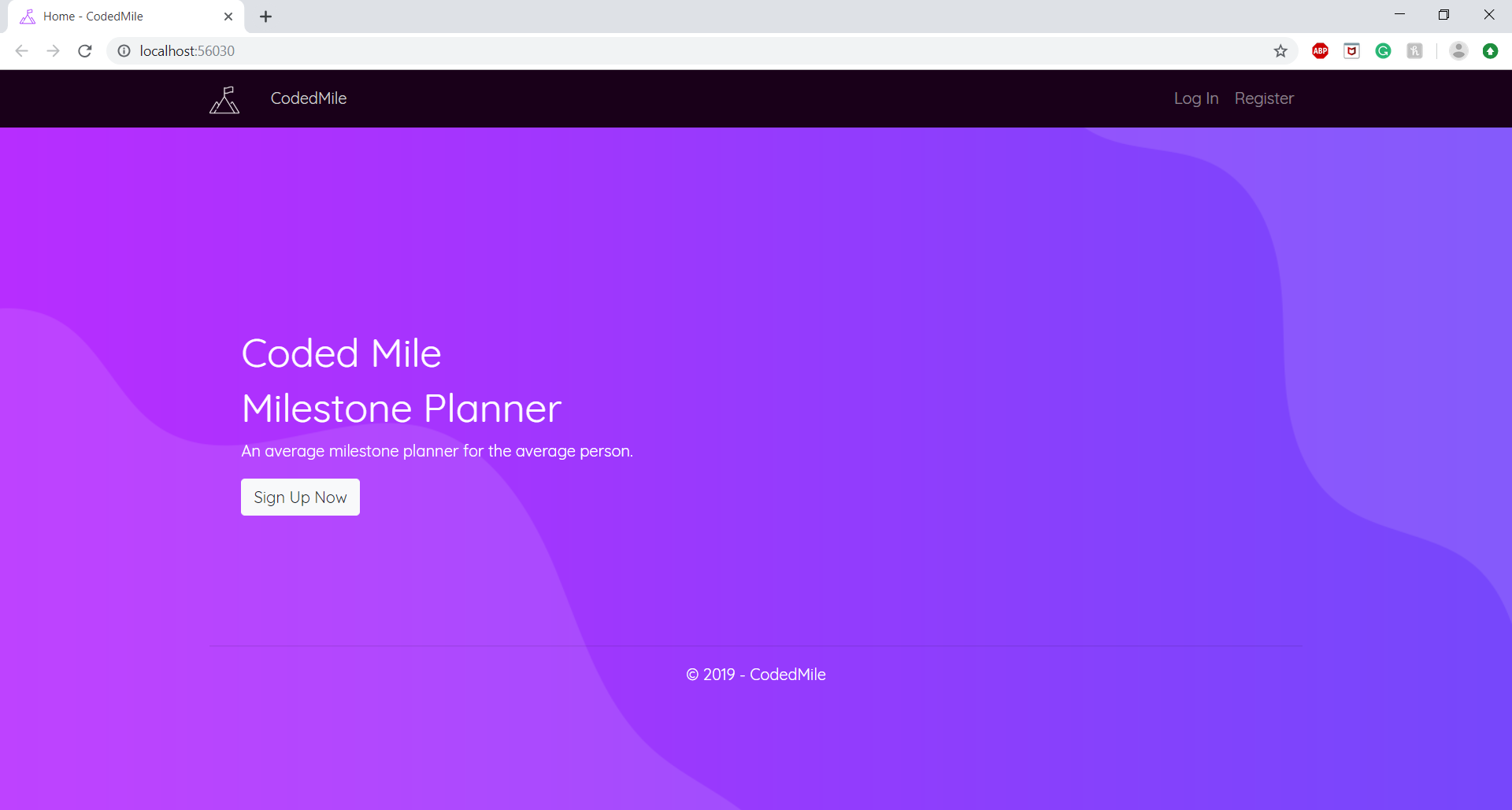


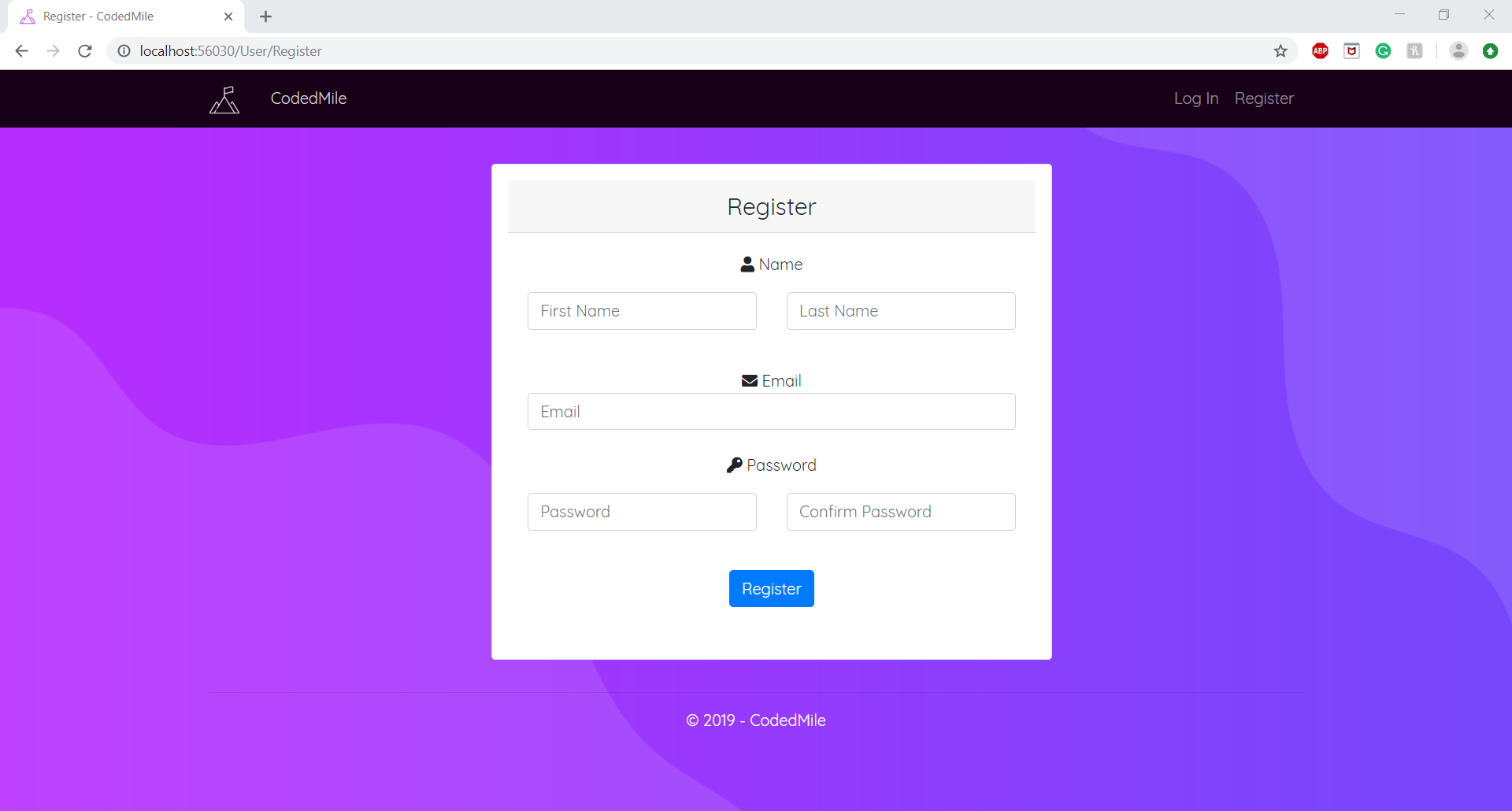
4.6)

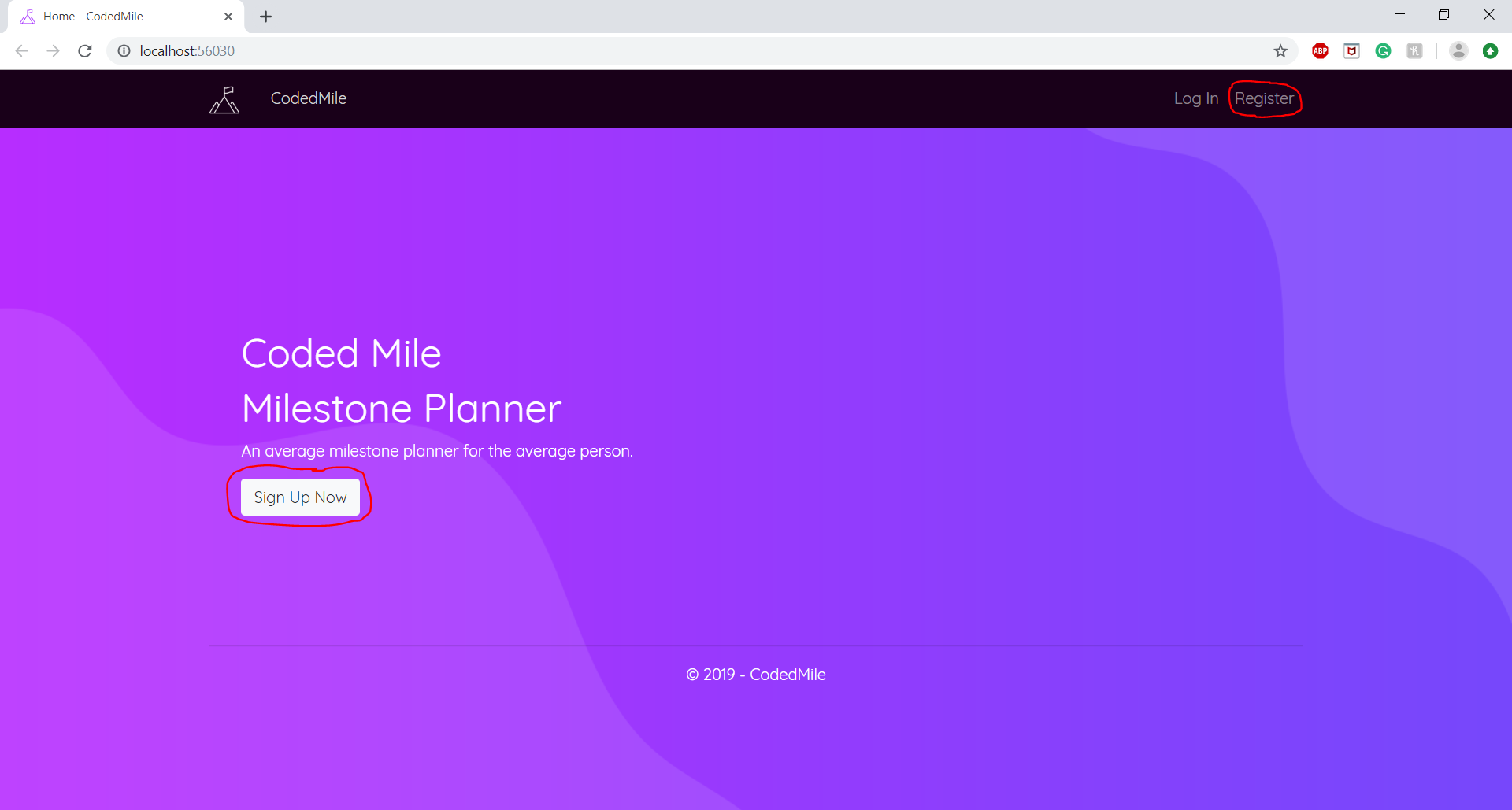


## Desktop

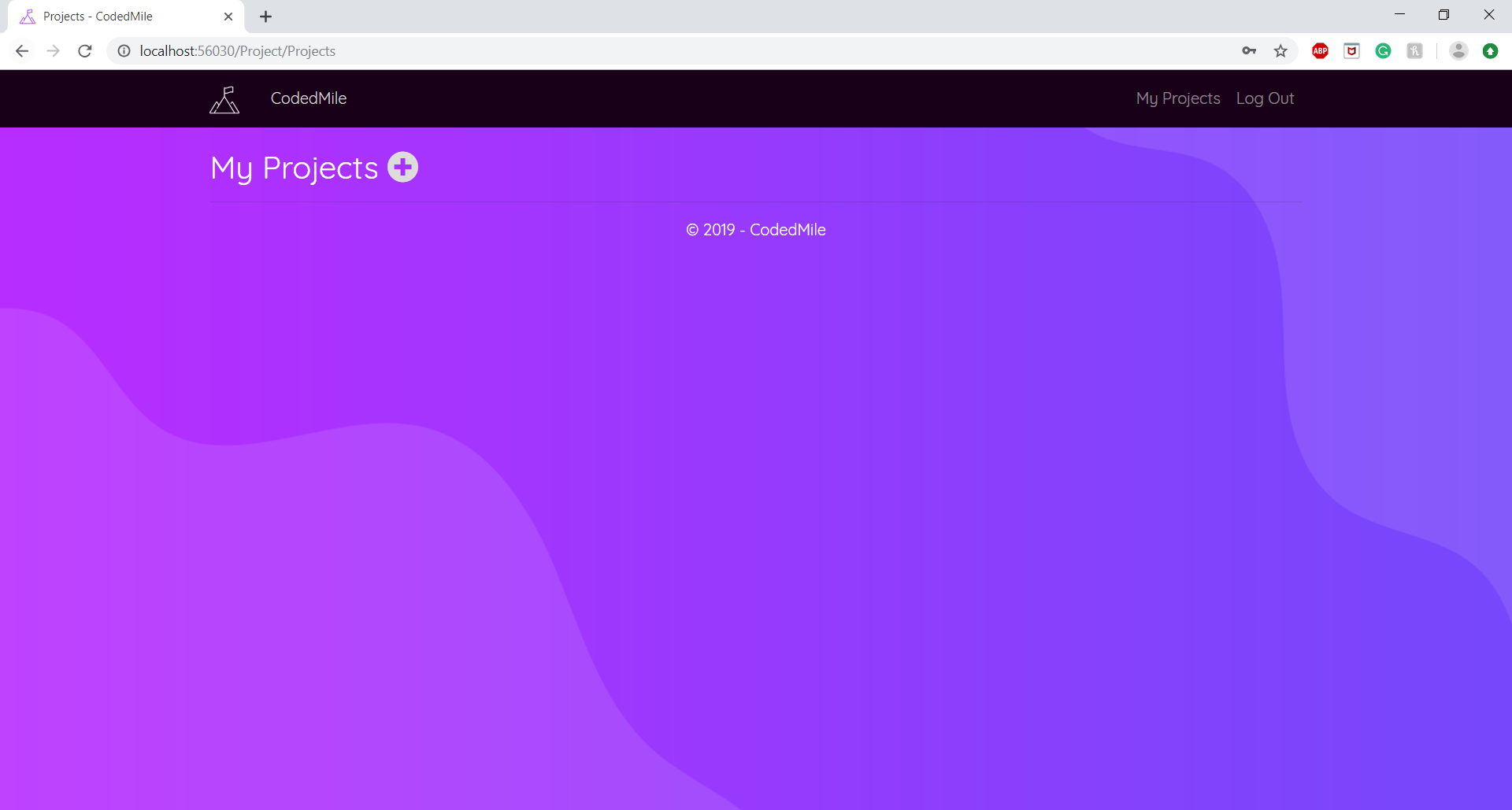
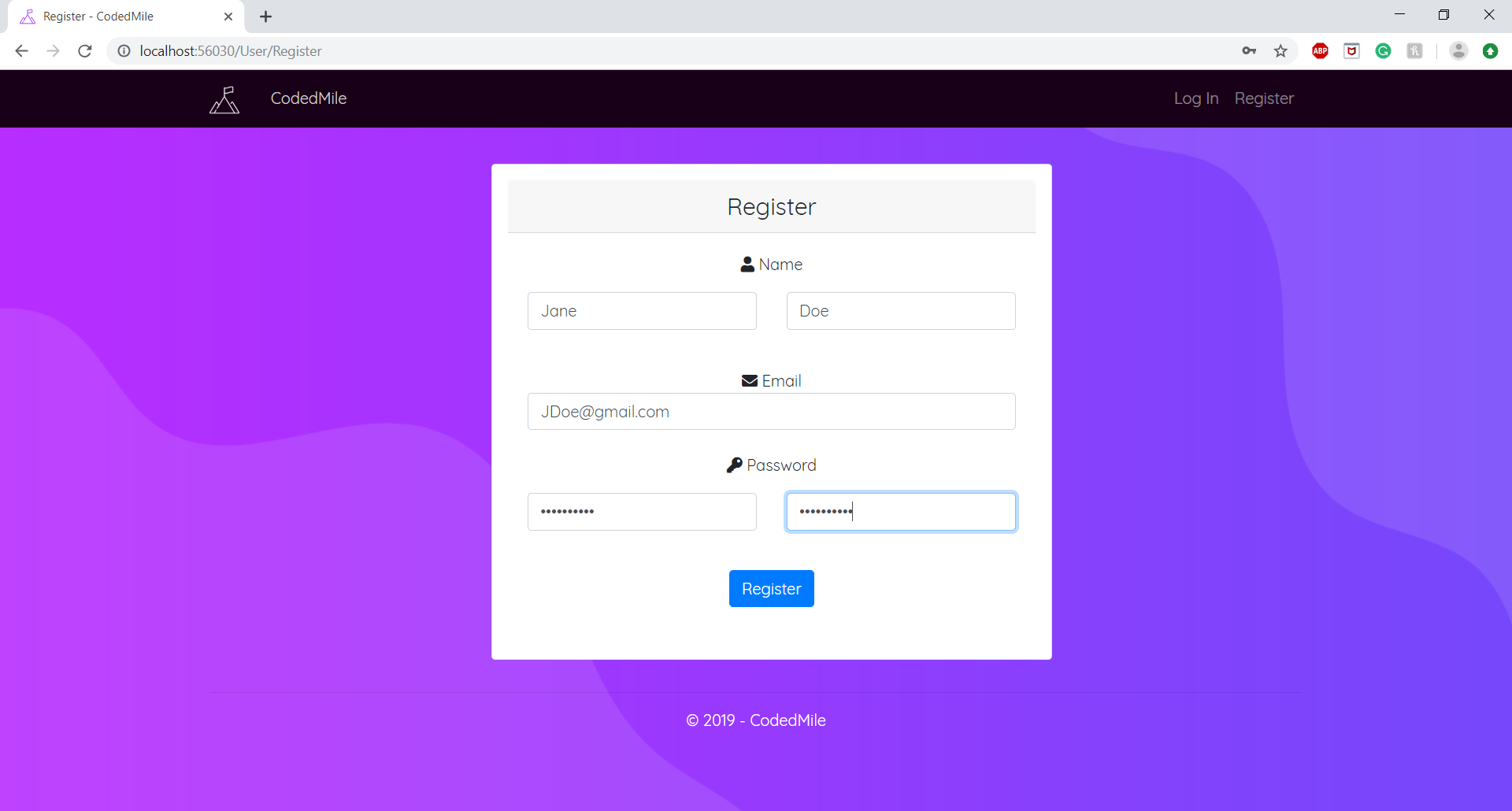
1



2



3



4

