***ASSIGNMENT NO. 2***

***Summary:***

***INTRODUCTION:***

Asynchronous code allows us to run multiple tasks concurrently on the same OS thread. In a typical threaded application, if you wanted to download two different webpages at the same time, async rust will do this work for you on the same OS thread and we can run multiple tasks at the same time on the same thread. In Async programming we can run the IO-bound at the same time on a single thread. After waiting for Some time, When we reach a point where we need the result of an asynchronous computation, we must .await it. In Rust, these values are known as futures.

***CREATION OF FUTURE:***

Async rust allow us to use future but it do not required the return type for the future.

let neg = negate\_async(1); This is how we can create the future of the negate\_async.

When our 1st task is completed then spawn will not let our program to execute till our 2nd is command is completed then our code will start its execution and then termination.

***EXECUTERS:***

In async rust we have executers but we need an external library to run executers and futures as the standard library does not come with an executor.

***ASYNC AWAIT:***

To use async await external library should be imported because rust’s standard library do not support or does not come with an executor so we need to reach out to an external crate for this.

In Rust, you need a dedicated executor. The executor is what takes care of executing the futures, polling them and returning the results when they’re done.

Async programming is very useful in Rust. We can learn more from it by exploring and trying to code more by ourselves.