**Version Control System**

**Class Number:**

**Project Name:** Version Control System

**Section :** CECS 543

**Team Name:** KARS

**Team Members:** Shraddha Nagargoje,Saumya, Anirudhha Shinde, Rucha Kanse , keval Goda

This application is a repository management system for local workspaces. Features supported by this application are:

* Create repository
* Checkin
* Checkout
* Listing manifest files
* Labelling manifest files
* Merge in
* Merge out
* Grandma

**Installation/Setup Steps:**

**Pre Requisites:** NodeJS should be installed on the system. Please refer the link for installation <https://nodesource.com/blog/installing-nodejs-tutorial-windows/>

**Build, Installation and Setup:**

Refer the following to build and run the project:

1. Extract the zip folder

2. Open command prompt and change directory to project path

a. cd ../../543-p1-KARS\_NEW/VCS

3. Install Express package- npm install express

npm install express

npm install fs-extra

npm install mkdir@latest

npm install dircompare

4. Execute the following commands:

a. node app.js

5. Open browser and type localhost:3000

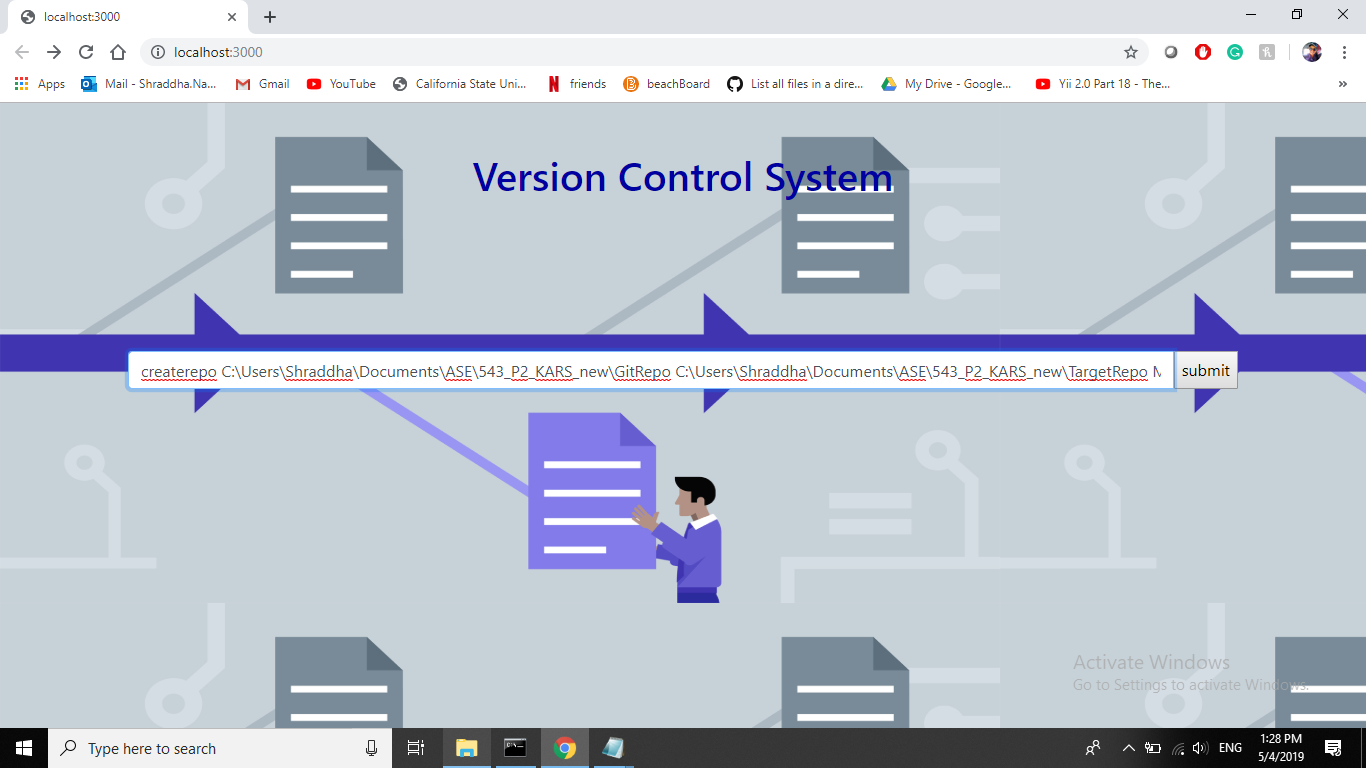
For all commands refer README File.

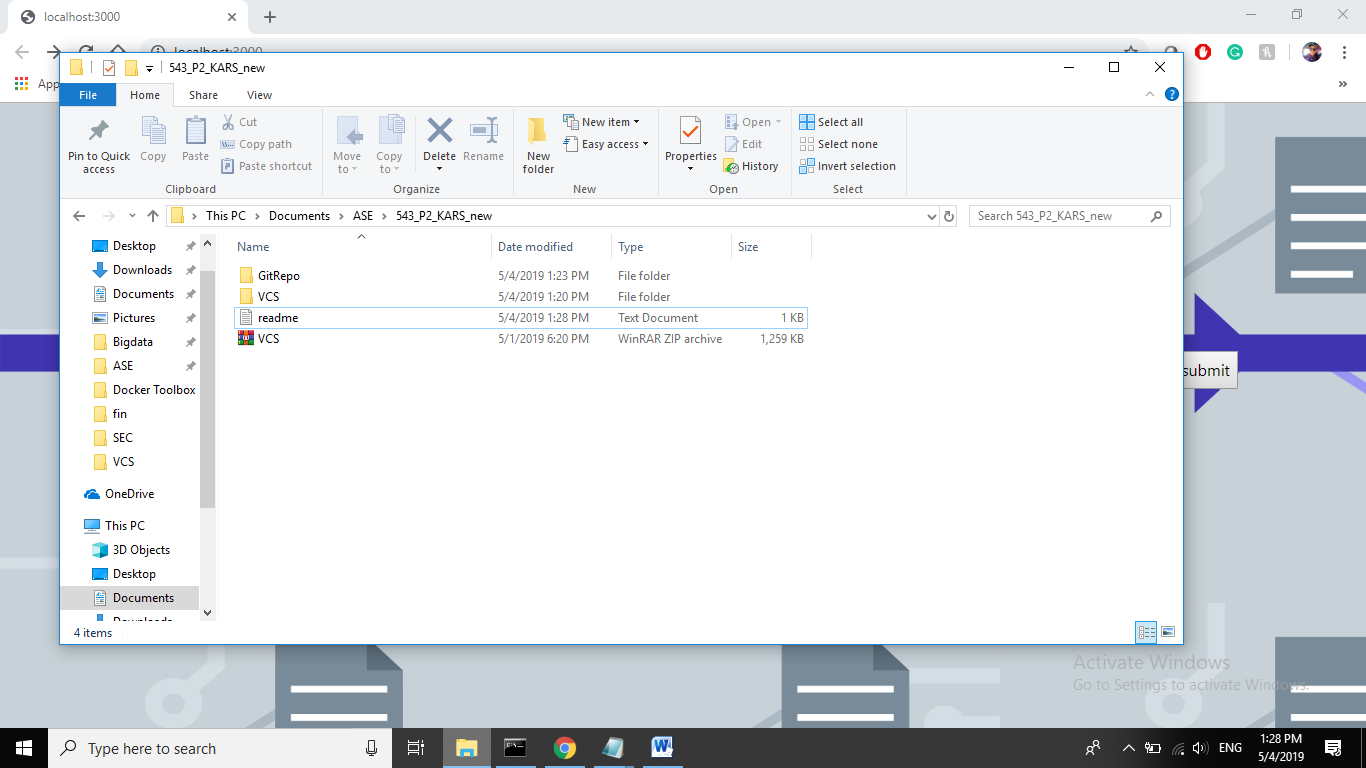
**Test Cases:**

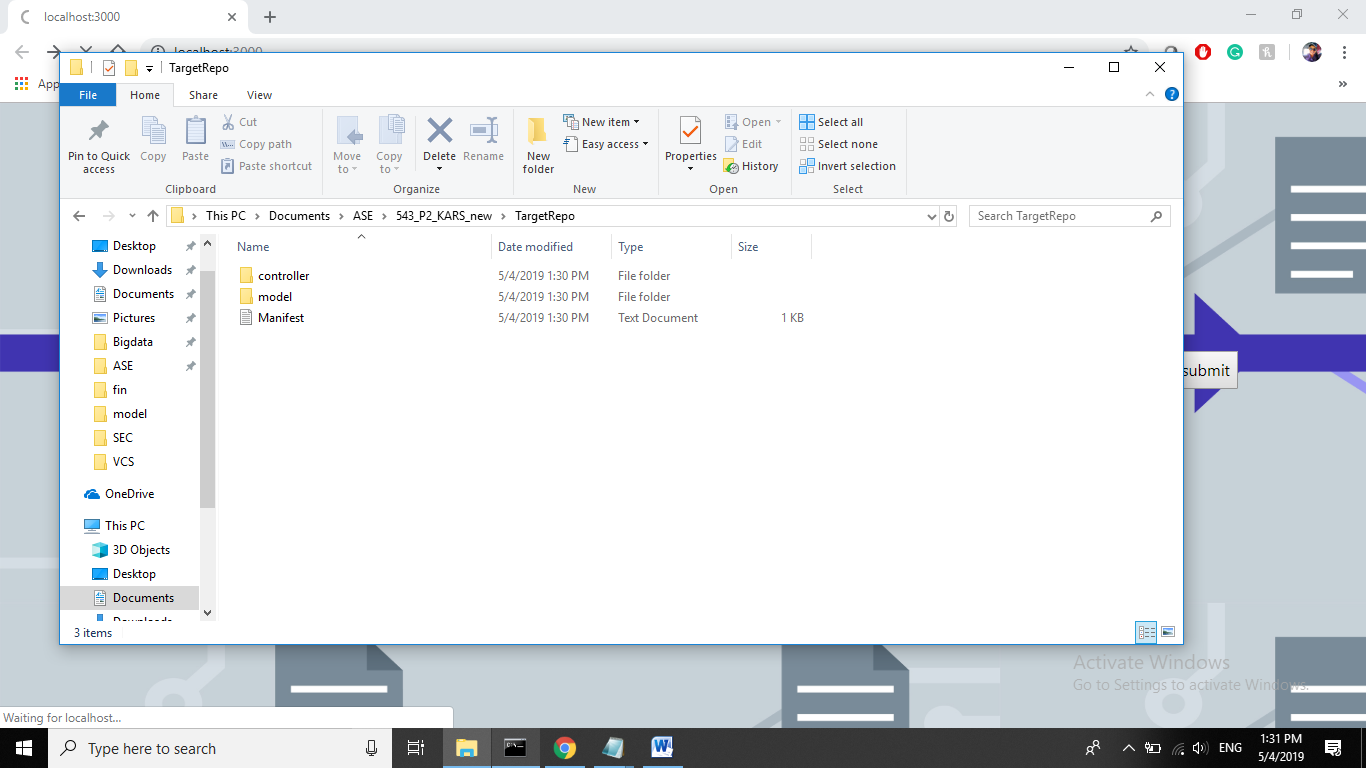
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr.No** | **TC\_Name** | **Objective** | **Pre-requisite** | **Description** | **Expected**  **Result** | **Actual Result** | **Status** |
| 1 | TC\_01 open application | Verify Home page is loaded. | Execute nodejs command. | Execute node app.js command and check whether local port is listening. | Home page should load on localhost:3000 | Nodejs is installed on the system. | Pass |
| 2 | TC\_02 Verify Home Page | Verify all User Interface components | Home Page is loaded. | Check whether all textbox and buttons are displayed on the Home Page appropriately | All User Interface Components are loaded on Home Page | Home Page is loaded successfully. | Pass |
| 3 | TC\_03 Create Repo | Verify Repository is created in given folder | Execute createrepo command | User should give createrepo<given\_folder\_name> command in the text box on the HomePage | Repository should be created in the given target folder. | Repository is created in the given target folder. | Pass |
| 4 | TC\_04 Create Manifest | Verify Manifest file is created. | Execute createrepo command | Manifest File should be created under target folder. | Manifest File should be created under target folder. | Snapshot summary is created under given target folder. | Pass |
| 5 | TC\_05 Manifest Content | Verify Manifest Content. | Refer TC\_04 | Manifest file should contain date and time stamp, command and path of target folder. | Manifest content should be appropriate. | Manifest content is appropriate. | Pass |
| 6 | TC\_06 Label Command | Verify Label of a Manifest file | User should provide a label for Manifest file. | Manifest file should be renamed according to the label given by the user. | Manifest file should be renamed. | Manifest file is renamed. | Pass |
| 7 | TC\_07 Checkout Command | Verify project is checked out | User provide a checkout command. | User will be able to recreate the new version of the project tree by selecting a manifest file in the repo. | Project should be checked out in the given target folder. | Project is checked out in the given target folder. | Pass |
| 8 | TC\_08 Checkin Command | Verify project is checked in. | User provide a checkin command. | User will be able to update the repository with a new version of the project. | Project should be updated with a new version. | Project is updated with a new version. | Pass |
| 9 | TC\_09 Listing | Display Manifest files. | User should provide the folder name | User will get a list of existing Manifest file with their labels. | List should be displayed on console. | List is displayed on console. | Pass |
| 10 | TC\_10 Mergeout Command | Verify merging of source and target project tree. | Execute mergeout command. | Source repo and target project tree should be compared and mismatched file should be placed in target folder with appropriate labels.(\_MR, \_MT) | 1.Target folder should include new files which are present in source that don’t match with target files with suffix \_MR.  2.Target folder should include new files which are missing from source with suffix \_MT.  3.Mergeout Manifest file should be created. | Target Folder included missing files from source and new files from target with appropriate suffixes.  2.Mergeout Manifest file created with date and time stamp and command. | Pass |
| 11 | TC\_11 Grandma | Grandma snapshot should be created. |  | Grandma snapshot should be created with all files with own suffix \_MG | Grandma snapshot should be created. | Grandma is created. | Pass |
| 12 | TC\_12 Mergein | Verify merge-in command | Execute merge-in command | Target Repository should be updated with new changed files. | Repository should be updated with new changed files. | Repository is updated. | Pass |

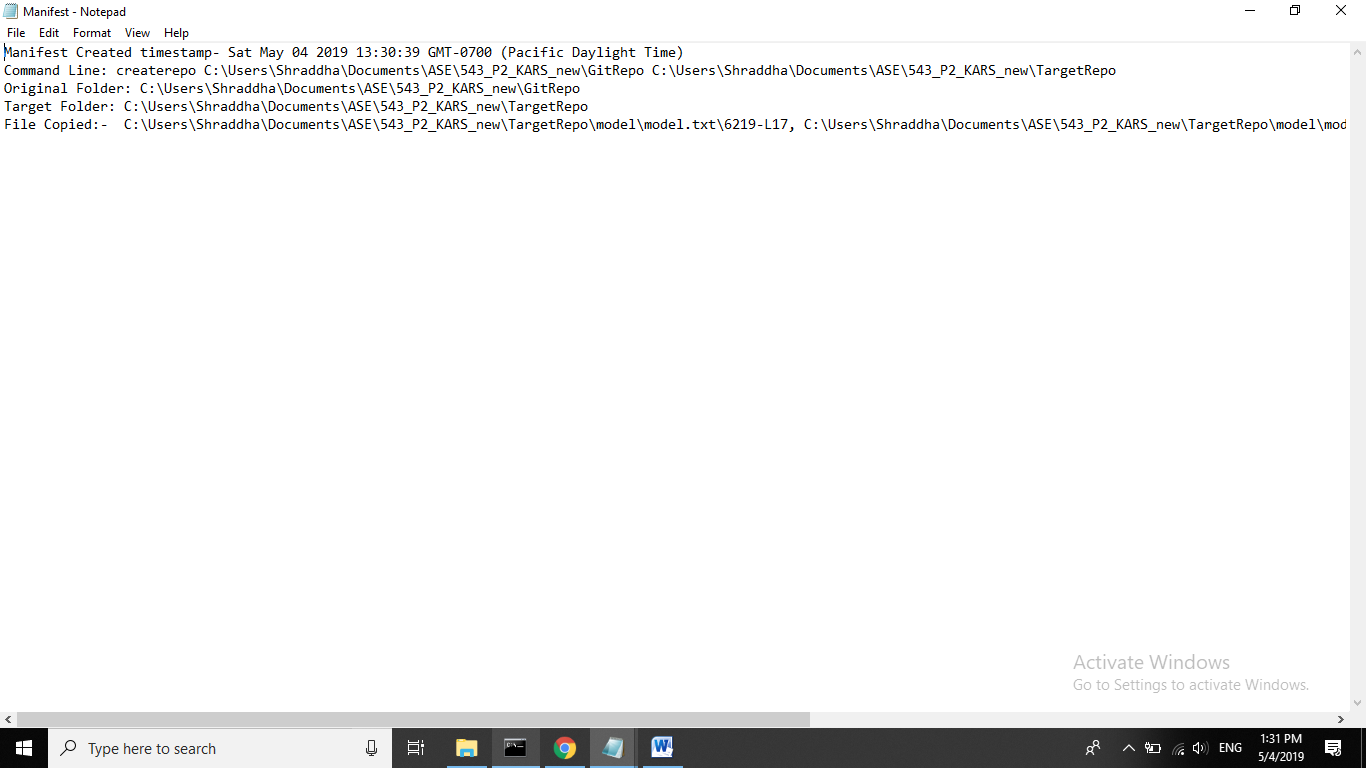
**Application Results:**

createrepo C:\Users \Documents\ASE\543\_P2\_KARS\_new\GitRepo C:\Users\ \Documents\ASE\543\_P2\_KARS\_new\TargetRepo Manifest



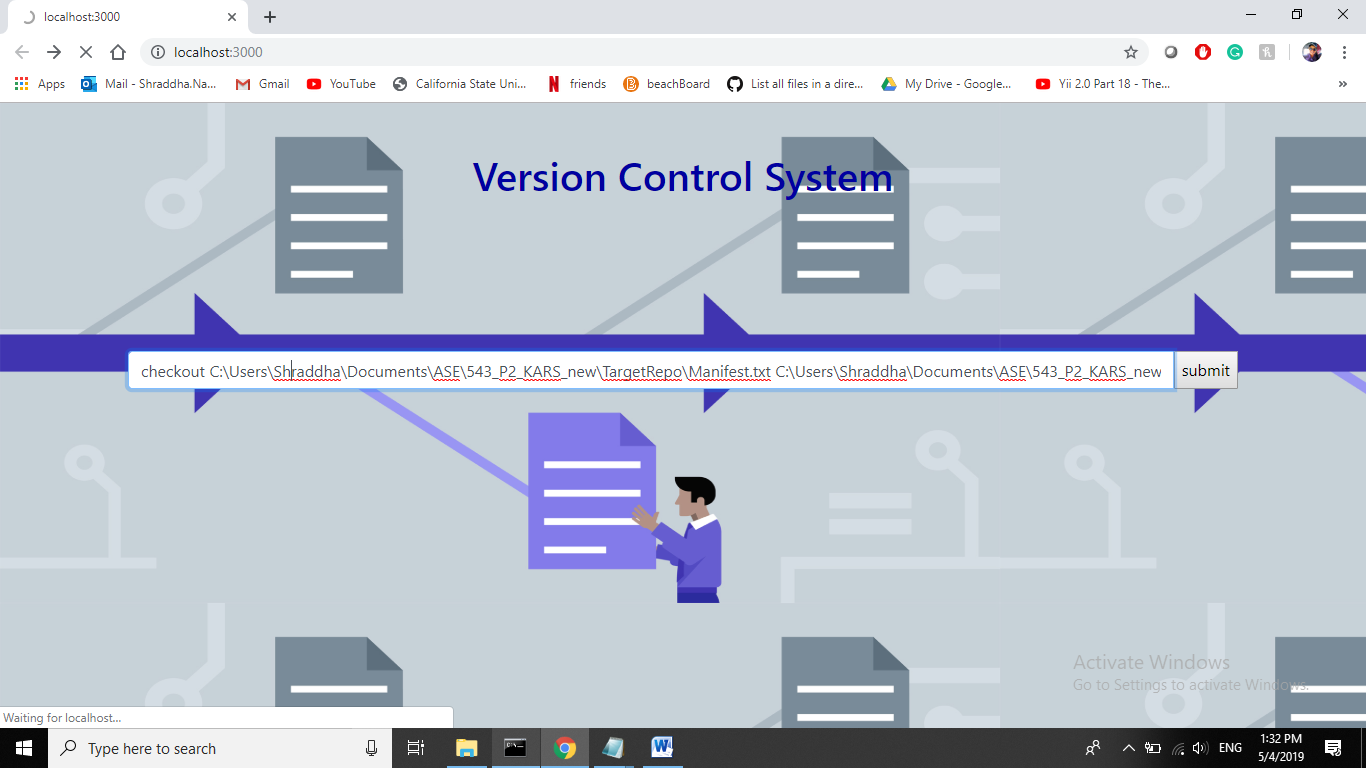


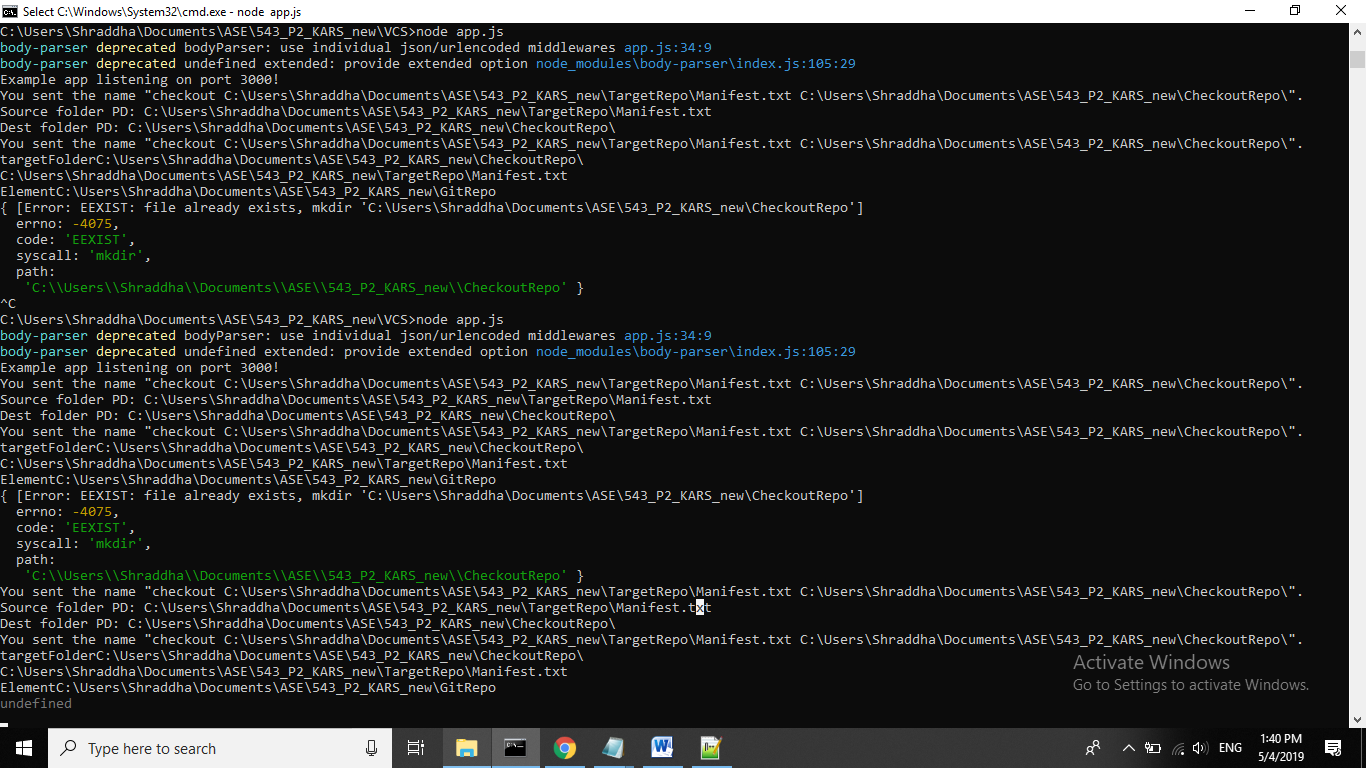


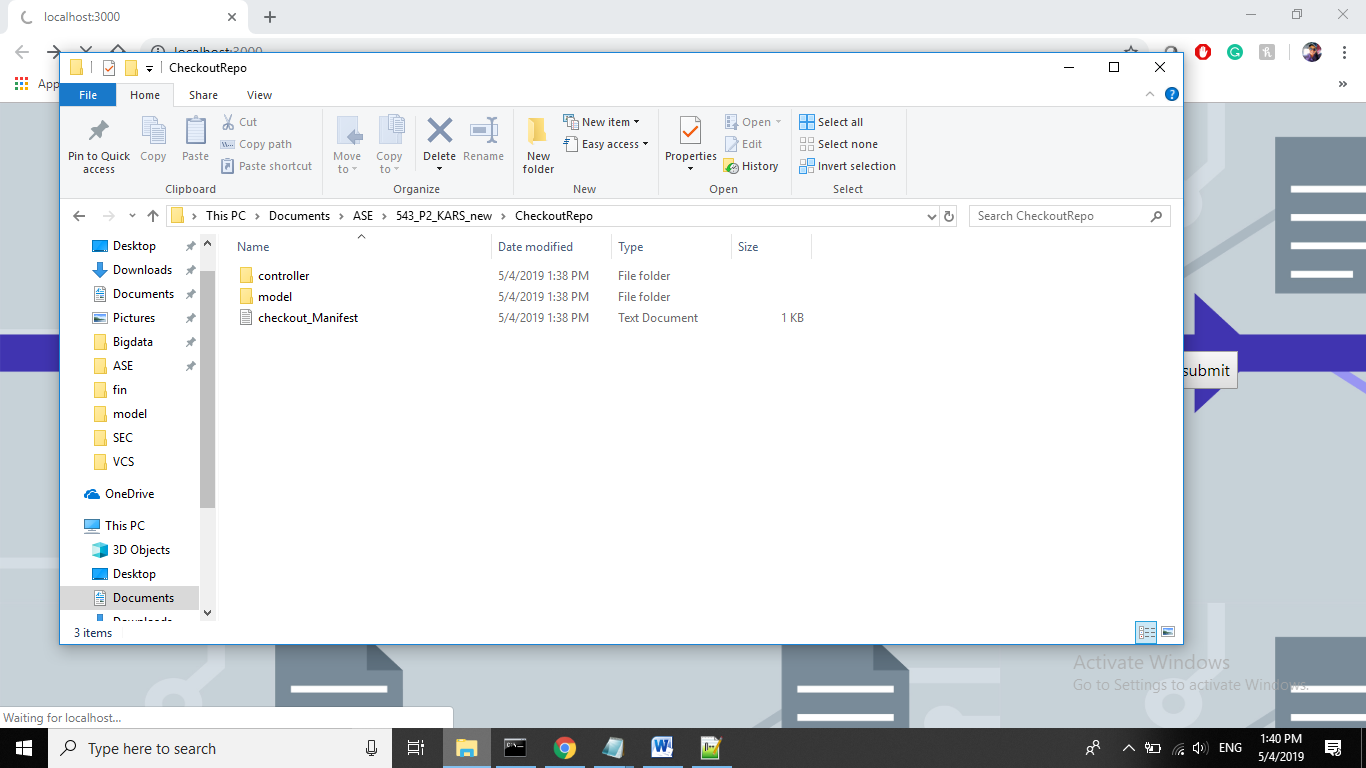


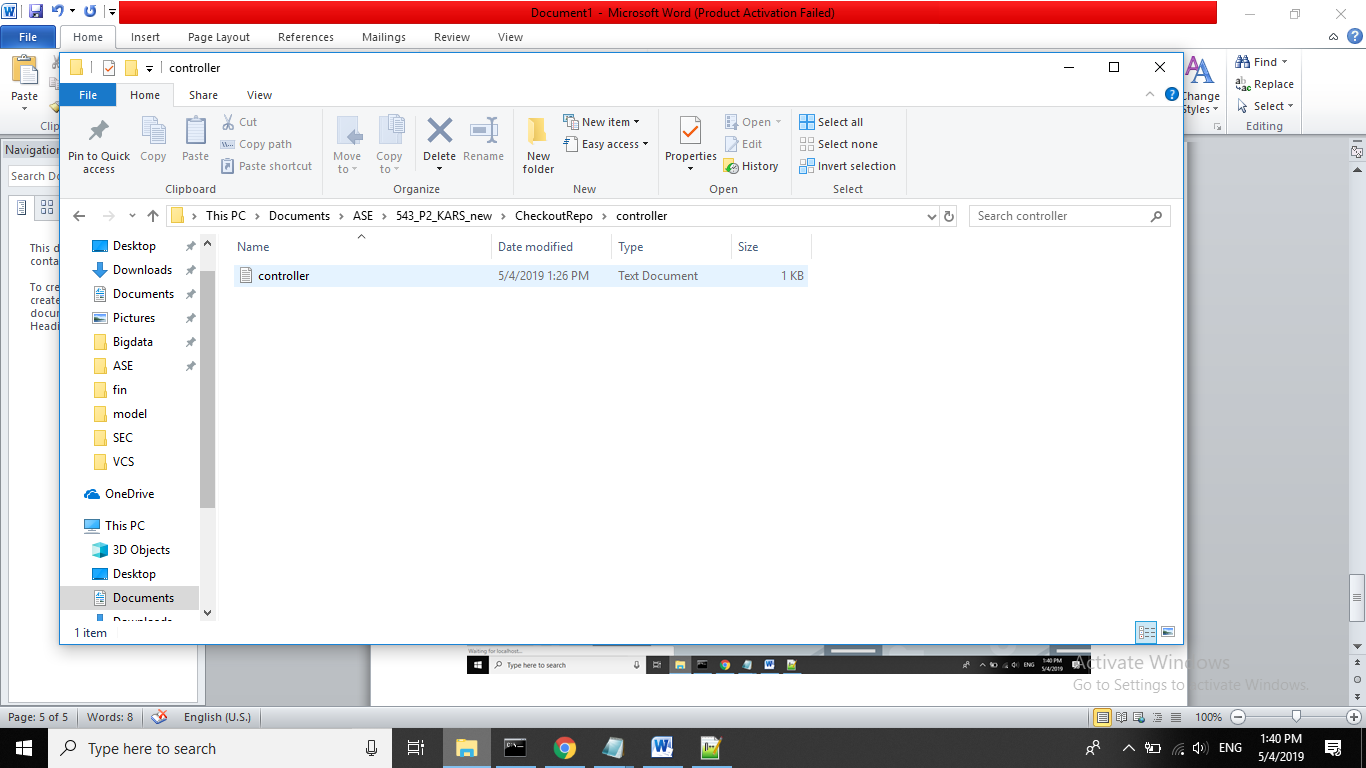
**Checkout**

checkout C:\Users\Documents\ASE\543\_P2\_KARS\_new\TargetRepo\Manifest.txt C:\Users \Documents\ASE\543\_P2\_KARS\_new\CheckoutRepo





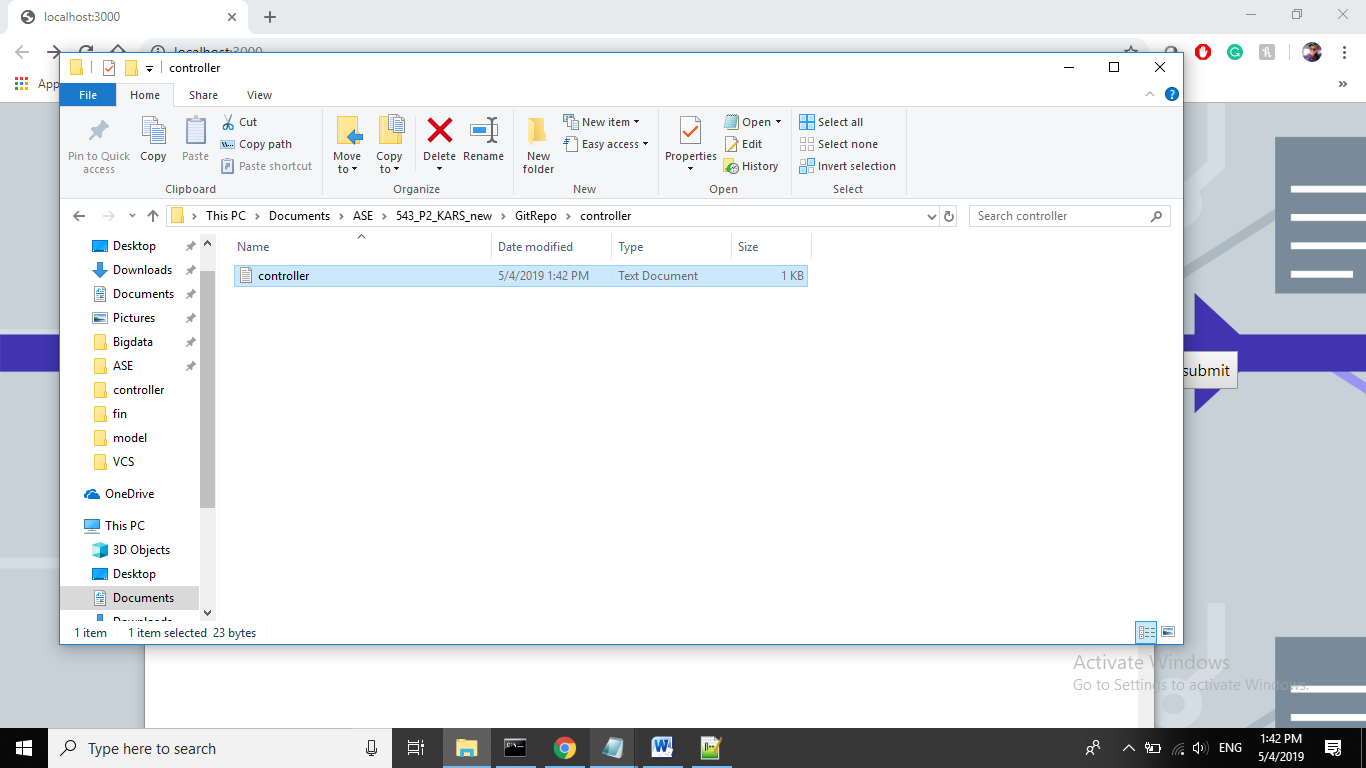


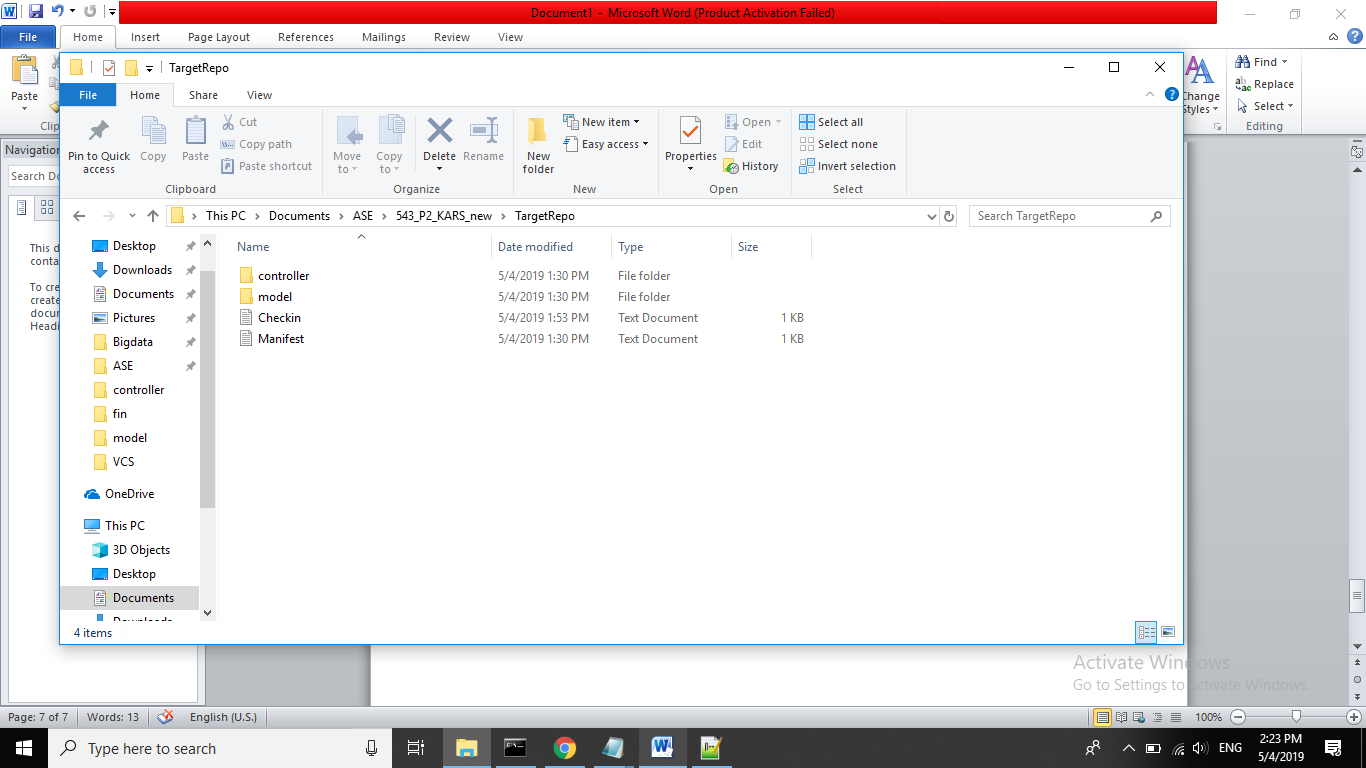


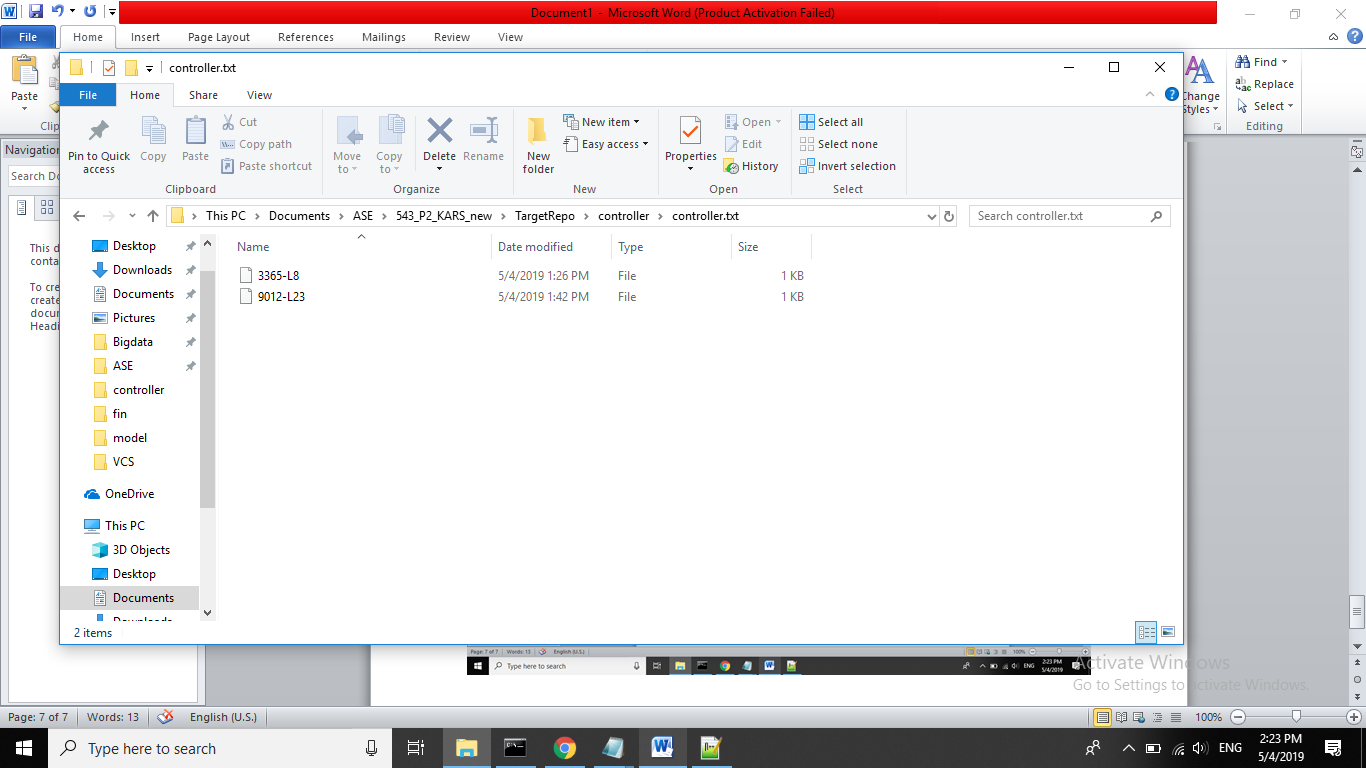


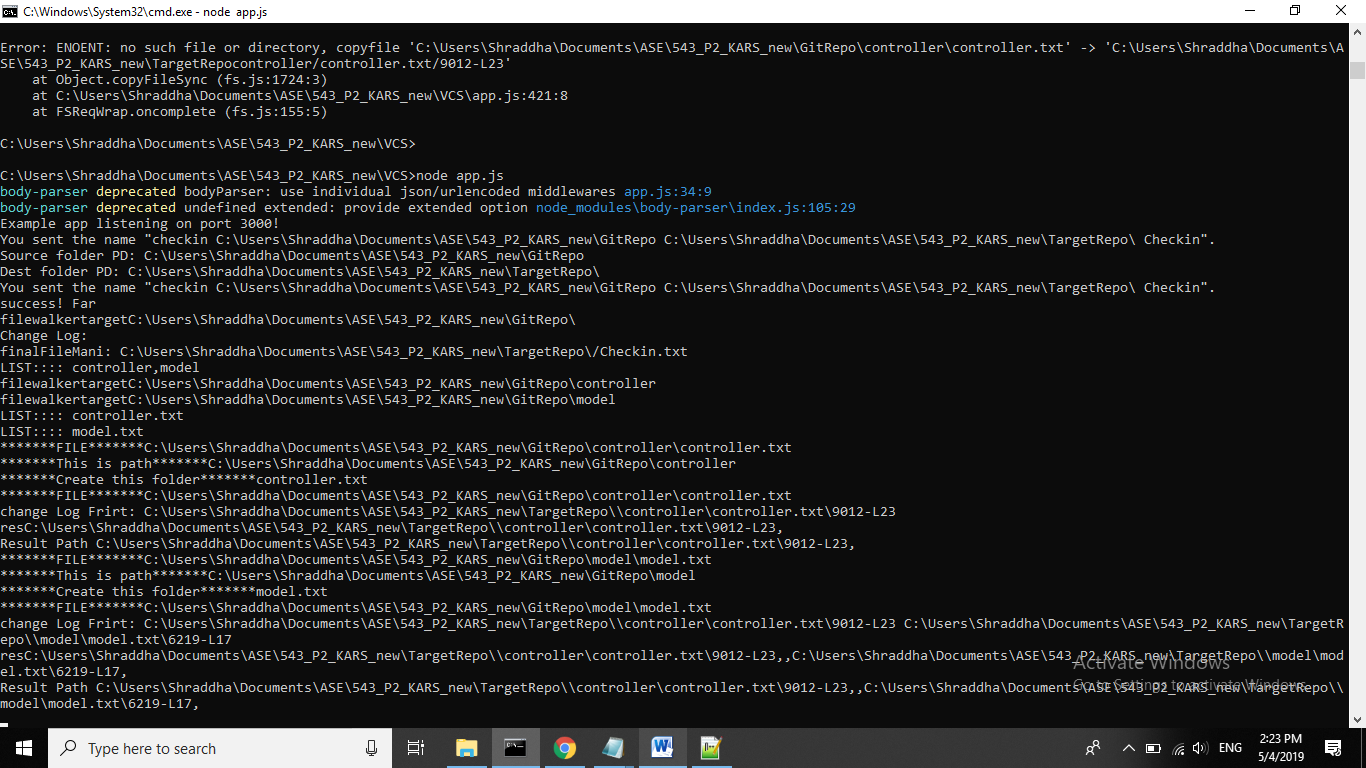
**Check-in**

checkin C:\Users\Documents\ASE\543\_P2\_KARS\_new\GitRepo C:\Users\Documents\ASE\543\_P2\_KARS\_new\TargetRepo\ Checkin

**Don’t forget to push slash(\) after targetRepo**







**Merge-out**

mergeout C:\Users\Documents\ASE\543\_P2\_KARS\_new\GitRepo C:\Users\Documents\ASE\543\_P2\_KARS\_new\GitRepoNewMerge

