**GRADUATE DIRECTED PROJECT**

**Project Name: Codeword**

**Team Name: Codeword Vue.js**

**Team Members:**

Naveen Kumar Chandaluri

Shivani Reddy Dodla

Sravya Kancharla

Sreelekha Vijaya

Srimai Reddy Yanala

Vyshnavi Yalamareddy

**Table of Contents**

**Page No**

**1.** **Introduction 3**

**1.1 Problem Statement**

**1.2 Proposed Solution**

**2.** **Methodology 3**

**3. Requirements 4**

**3.1 Functional Requirements**

**3.2 Non-Functional Requirements**

**4. Design 7**

**5. ER-Diagram 8**

**6. Dummy Data 9**

**7. Tools and Technologies 16**

**8. Developer Manual 35**

**9. Test Cases 44**

**10. Results and analysis of test cases 44**

**11. Deployment 47**

**1. Introduction:**

Codeword involves creating a progressive web app that assigns confidential codewords to each student in a course. Instead of printing out a separate codeword on a paper and distributing it among the students, the process is automated which will assign the codewords for all the students in a particular course. In general, code words are typically used to maintain confidentiality and reliability. Students will be assigned each codeword when they register for the course.

**1.1 Problem Statement:**

According to the client, distributing codeword to students are being done manually since many days. At the beginning of the semester, student is handed out with a codeword on a paper. Student needs to enter the codeword to take up a survey related to the course. Once the course is completed, student needs to take post survey where they need to enter the same codeword which they used for the pre survey.. There might be a chance of misplacing the paper with the codeword and a scenario where student forgets the codeword which results in not taking up the survey at the end of the course. This process involves unnecessary use of time and resources.

**1.2 Proposed Solution:**

To minimize the problems that the client is facing, we have come up with a solution where distributing the codewords was made easy which doesn’t involve paper. We named it as “ Codeword ” which displays codewords for each registered course. In this application, instructor has an option to add the student and course details which will be displayed on the dashboard. Also has an ability to add course with the start and end date. Using this application, user doesn’t have to remember the codeword as it is available on the website for the registered course.

**2 Methodology:**

In this project, we have used agile methodology for software development. Throughout the development process, this methodology ensures that the value is optimized. We have used iterative planning and feedback results to align a delivered product which reflected the needs of our client. In this process, we have adapted to change requirements by evaluating and measuring the status of a project. Agile provides a unique opportunity for clients to be involved during the project review sessions for new feature. We have fixed schedule of sprints of weekly or every 15 days where new features are delivered with a high level of positive predictability. We also had regular stand up meetings where we discussed about During the development of the software, this methodology encourages collaboration between the client and the team which offers mutual benefits in the mitigation of the high risks. Without the concern for any hierarchy or team member roles, solutions and requirements evolve through cooperation between the self organized teams. This methodology promotes teamwork, collaboration and process adaptability throughout the project.

**3. Requirements**

**3.1 Functional Requirements**

1. As a user, I want to register by providing my email ID, so that I will get an email with a system generated password.
2. As a user, I want to log in with my Email Id and password.
3. As a user, if I forget the password, I want to retrieve my account by entering the registered Email ID and submit it. Later I can access my account with the temporary password provided.
4. As an instructor, I want to check an "Instructor?" box when I register so that I can distribute code words.
5. As an instructor, I want to see my list of courses, those are “active” today between the start and end date inclusively.
6. As an instructor, I want to check an "Inactive" box to see completed courses.
7. As a user, I want to navigate between multiple pages (Home (list of active courses), + New Course, View a Set of Codewords and Confirm Delete Course).
8. As an instructor, I want to create a new set of code words by entering a unique name for the codeword set and uploading a plain text file that contains a new list of codewords.
9. As an instructor, I want to create a new course by adding a list of students (containing one \*@ email address & one name field for each student), selecting a codeword set from a drop-down that shows the name of the set with the calculated count in parenthesis, and setting the start date (default to today) and end date (defaulted to 4 Months after then start date) for the course, and a pre-survey URL, and a post-survey URL.
10. As an instructor, if I click on “View Students” I want to see a list of students for the selected course (emails and names) and want to see the calculated count of students in that course.
11. As an instructor, when I view my list of active courses, for each active course, I want to see:
12. The unique name
13. The codeword set name with the calculated count of words
14. The start date and the end date
15. The pre-survey URL and the post-survey URL
16. An option to “Edit”
17. An option to “Delete” an active course (expired courses will be not appear).
18. As an instructor, I want to delete a student from an active course (e.g. when they drop).
19. As an instructor, I want to add a student to an active course (e.g. when a new student transfers in)
20. The unique email
21. The student name
22. As an instructor, I want to edit a course by clicking "Edit" on a course entry. I want to be able to modify:
23. Student name
24. Student email
25. The pre-survey URL and the post-survey URL
26. Start date and end date.
27. As an instructor, I want to delete an active course, by clicking the delete button on the view courses screen, getting a confirmation box that says" Are you sure you want to delete course ? This action is not reversible. with two buttons "Delete", "Cancel". If I click cancel, I return to the view courses screen, if I click "Delete", the course will be deleted (along with the course student list).
28. As an instructor, I want to delete an existing course once it is completed.
29. As an instructor, I want one codeword from the specified set to be randomly assigned to only one student and each student must have a unique codeword in that course and each codeword can only be issued once in each course (or not used at all) when a course is created.
30. As an instructor, I want the app to always have my Basic small codeword set (with 90 unique codewords) and a second Large Codeword Set (with 200 unique codewords) and instructors are not allowed to delete them.
31. As an instructor, I want to be notified if the codeword set is too small. If the calculated count of codewords is less than the calculated count of students then I want to see a warning that says “You have n students, but the codeword set has only m words.
32. As an instructor, I want the codewords confidential and must not know or be able to see the code word each student gets.
33. As an instructor, I want to edit a codeword by clicking "Edit" .
34. As an instructor, I want to add a new codeword to an existing codeword set
35. The codeword should be unique
36. The codeword should contain only alphabets.
37. The codeword should contain minimum 5 alphabets.
38. As an instructor, I want the percentage of students who have accessed their codewords in that course.
39. As an instructor, I want to search a student courses or codeword in a codeword set in a search bar.
40. As a student, I want to see my courses and see/acknowledge/get my random codeword for the respective course on my home page.
41. As a student, I want to do pre-survey URL and the post-survey URL when available.

**Non-Functional Requirements**

1. Each course is limited to no more than 1000 students.
2. As an Instructor, while creating codeword set, If there is more than 1200 codewords in the file which is getting uploaded to create codeword set, Instructor should be notified with a pop up message that "Each codeword set has a limit of upto 1200 codewords".
3. As an instructor, I want to see the calculated count of students for codewords when I view a course.
4. As an instructor, I want to see less than 10 active courses.
5. As a User, I want Email id to be auto-filled.

**Design:**

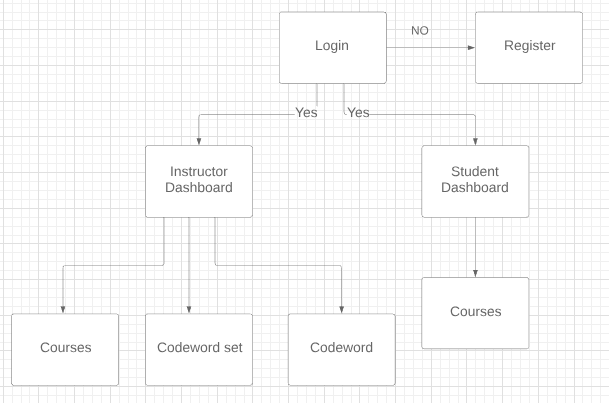
****

Figure 1

Block Diagram

**ER Diagram:**

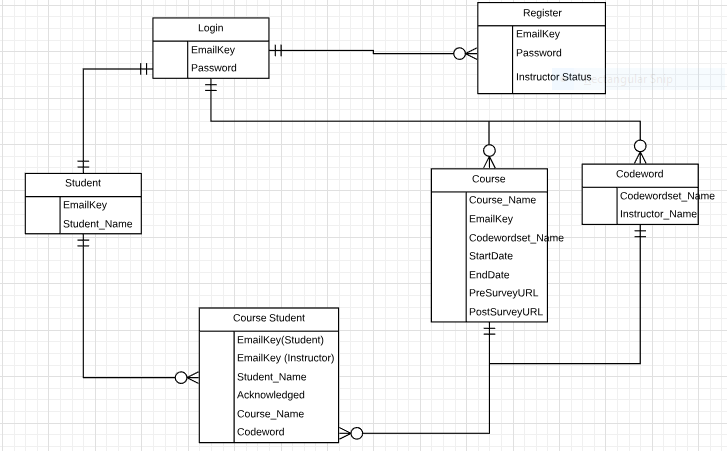
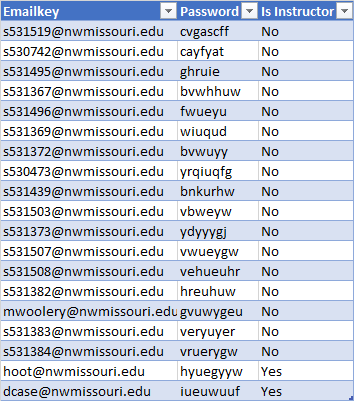
****

Figure 2

ER Diagram

**Dummy Data:**

**Table Login:**

****

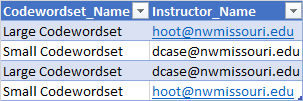
**Student Table:**

|  |  |
| --- | --- |
| **EmailKey** | **Student\_Name** |
| **s531519@nwmissouri.edu** | **Yashwanth Reddy Bommineni** |
| **s530742@nwmissouri.edu** | **Naveen Kumar Chandaluri** |
| **s531495@nwmissouri.edu** | **Chaithanya Cherukuru** |
| **s531367@nwmissouri.edu** | **Sai Sirisha Devineni** |
| **s531496@nwmissouri.edu** | **Shivani Reddy Dodla** |
| **s531369@nwmissouri.edu** | **Girish Guntuku** |
| **s531372@nwmissouri.edu** | **Anurag Kumar** |
| **s530473@nwmissouri.edu** | **Ujjawal Kumar** |
| **s531439@nwmissouri.edu** | **Dattu Bhargav Medarametla** |
| **s531503@nwmissouri.edu** | **Santosh Sekhar Muchkur Bogarajula** |
| **s531373@nwmissouri.edu** | **Naveenkumar Nuggu** |
| **s531507@nwmissouri.edu** | **Vijay Kumar Tupakula** |
| **s531508@nwmissouri.edu** | **Satya Sai Ram Vankina** |
| **s531382@nwmissouri.edu** | **Sreelekha Vijaya** |
| **mwoolery@nwmissouri.edu** | **Matthew Woolery** |
| **s531383@nwmissouri.edu** | **Vyshnavi Yalamareddy** |
| **s531384@nwmissouri.edu** | **Srimai Reddy Yanala** |
| **hoot@nwmissouri.edu** | **Charles Hoot** |
| **dcase@nwmissouri.edu** | **Denise Case** |

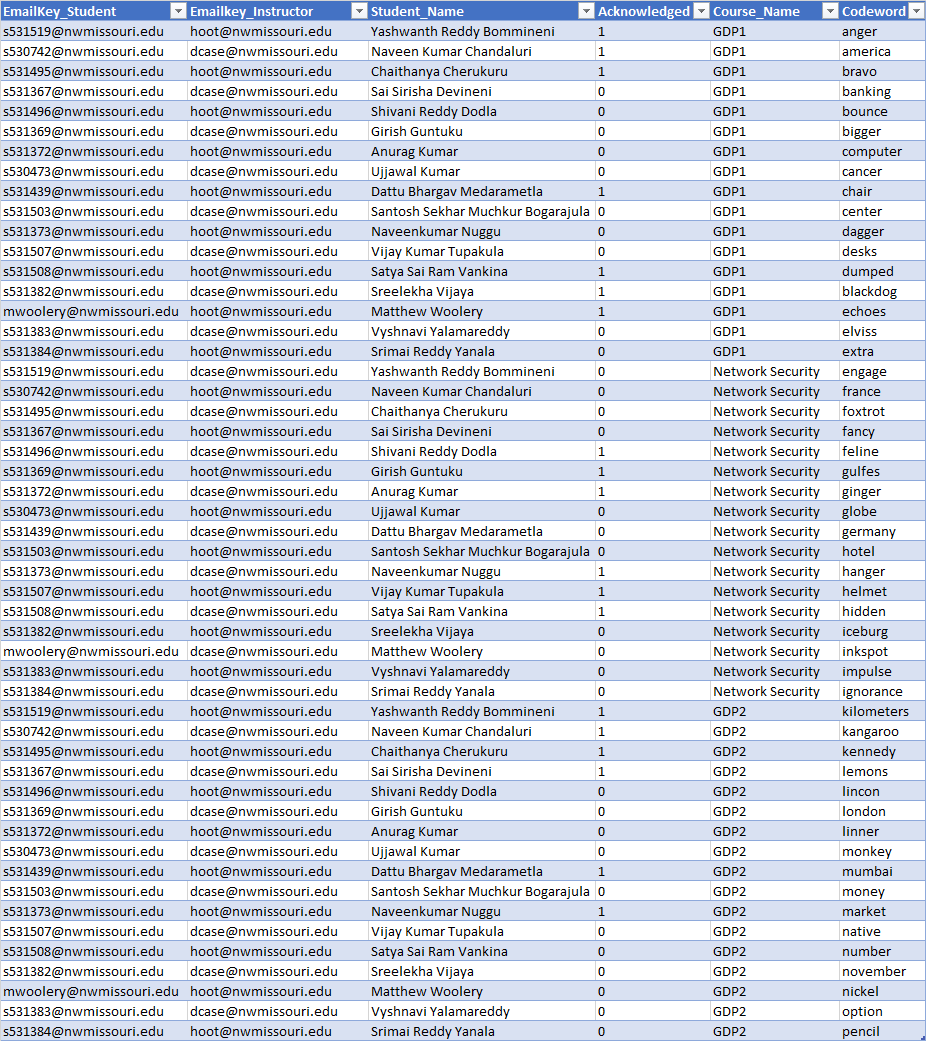
**Course Table:**

****

**Codeword Table:**

****

**CourseStudent Table:**

****

**Design pattern:**

Model, view and controller is an architectural pattern commonly used for developing user interfaces that divides an application into three interconnected parts.

**Model:**

The model represents the data and does not depend on the controller and view. It directly manages the data, logic and rules of the application. An instance of model is called a document and models are responsible for creating and reading documents from the MongoDB database.

**View:**

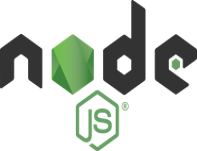
Displays data to user in easy to understand format based on the user’s actions. The view component is used for all the UI logic of the application.Vue.js and Bootstrap are used in front end as a view which present the model in a particular format.

**Controller:**

A controller is the link between user and the system. It controls the data flow into model object and updates the view whenever the data changes.Node.js is used in backend as a controller.

**Tools and Technology:**

**Block Diagram:**



Front-End

Back-End

1. Http Request

2. Database Request

3. Database Response

4. Http Response

5. Deployed

Figure 3

Tools and Technologies

Front end (vue.js, Bootstrap and axios) sends http request to the back end (Node.js), back-end sends database request to the mongo DB. MongoDB checks with the data and sends DB response to the back end, Front-end receives http response from back-end and this application deployed using heroku.

a) Vue.js:

Vue.js is one of the new software technologies that are being widely used across the world for web development. It is a JavaScript framework with various optional tools for building user interfaces.Vue.js is small in size, easy to understand and develop applications. It has detailed documentation and has a great deal of flexibility. Vue.js also facilitates two-way communications because of its MVVM architecture which makes it quite easy to handle HTML blocks.

B) Bootstrap:

Bootstrap is a free and open source front end web framework. It contains HTML and CSS based design templates for typography, forms, buttons, navigation and other interface components which save lots of time and easy to use. Using bootstrap can easily create responsive designs and it is open source where it is completely free to download and use.

c) Node.js:

Node.js is an open source, cross platform JavaScript run time environment that executes JavaScript code outside of a browser and JavaScript is used primarily for client-side scripting. It lets developers use JavaScript to write command line tools for server-side scripting, running scripts server side to produce dynamic web page content before the page is sent to the user’s web browser. It uses an event driven, on blocking I/O model that makes it lightweight and efficient.

d) MongoDB:

MongoDB is a cross-platform document-oriented database program. Classified as a NoSQL database program, it used JSON like documents with schemata. It stores data in flexible and it is a distributed database at its core, high availability, horizontal scaling and geographic distribution are built in and easy to use. Ad hoc queries, indexing and real time aggregation provide powerful ways to access and analyze the data and it is free to use.

e) MongoDB Atlas:

MongoDB Atlas is a fully-managed cloud database that build MongoDB. Atlas handles all the complexity of deploying, managing and healing your deployments on the cloud service.

F) Heroku:

Heroku is a cloud platform as a service (Paas) which supports several programming languages. It is one of the first cloud platforms and it is said to be a polyglot platform as it has features for a developer to build, run and scale applications in a similar manner across most languages. It eliminates the need for expensive infrastructure and make tremendous cost savings and also supports a growing number of coding languages.

G) GitHub:

GitHub is a web-based hosting service for version control, which is used for storing the modifications in a central repository. This allows developers to easily collaborate, as they can download a new version of the software, make changes and upload the newest version. Every developer can see these new changes, download them and contribute.

GitHub essentials are:

1) Repositories

2) Branches

3) Commits

4) pull requests

Repositories:

A GitHub repository can be used to store a development project, it can contain folders and any type of files (HTML, CSS, JavaScript). It also should include license file and README file about the project.

Branches:

A GitHub branch is used to work with different versions of a repository at the same time, by default a repository has a master branch and any other branch is a copy of the master branch. When changes are ready, they can be merged into the master branch.

Commits:

At GitHub changes are called commits. Each commit (change) has a description explaining why a change has made.

Pull Requests:

At GitHub pull requests tell about the changes we have pushed to a branch in a repository on GitHub. Once pull request is opened, we can discuss and review the potential changes with collaborators.

h) Visual Studio Code:

Visual Studio Code is a powerful source code editor which runs on desktop and is available for Windows, macOS and Linux. It is a half way between a text editor and an IDE and it comes with a built-in support for JavaScript, Typescript, and node.js. Visual Studio Code has the best typescript support and also has a great ecosystem of plugins for supporting other languages (C, C++, c#, Python…). It is cross platform which means that we don't have to learn another tool.

I) Postman:

Postman is a Google Chrome Plug-in for API development and testing. We can build, test and document our APIs faster, can share our work with other team members.

The main features of postman are

1) Powerful, Simple to use GUI

2) Unlimited collections, environment, tests and sharing.

3) Extensive team collaboration tools

4) Maintain history of API requests

5) Extended API documentation and monitoring features.

j) Mail gun:

Mail gun is an email automation service provided by Rackspace. It offers a complete cloud based email service for sending, receiving and tracking email sent through websites and applications. Mailgun features are available through an intuitive RESTful API or using traditional email protocols like SMTP. Mailgun is capable of integrating with a local postfix service, sending email through SMTP and integrating programmatically with existing cloud applications.

**Installation Guidelines:**

1. **Visual Studio Code**
   1. Download and install visual studio code from the below link<https://code.visualstudio.com/>

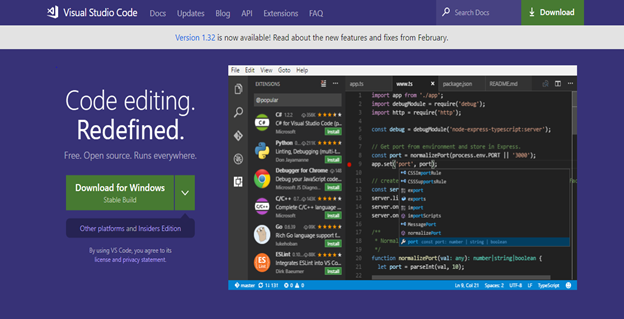


Figure 4

Download of the visual studio code

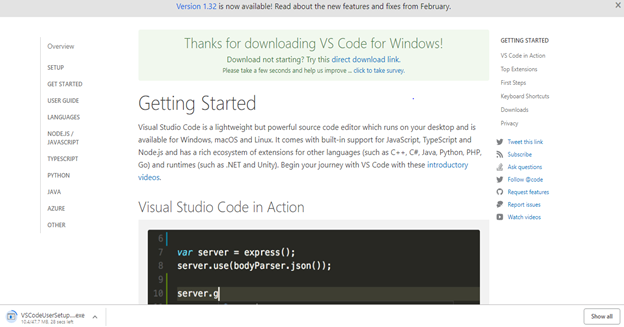


Figure 5

Getting started with the visual studio code

b) After downloading open the .exe file for installation. The Setup wizard appears.



Figure 6

Installation of the visual studio code

c) In setup wizard click on next and then accept the license agreement, click next to continue.

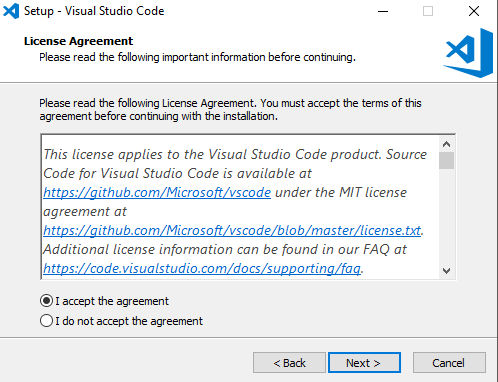


Figure 7

Continuation for the installation of the visual studio code

d) Give path where you want visual code to be installed

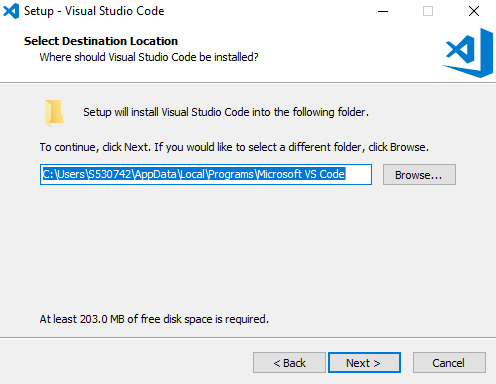


Figure 8

Path to be given for visual studio installation

e) Select if you want to create a Start Menu Folder for Visual Studio Code and then to proceed click on Next.

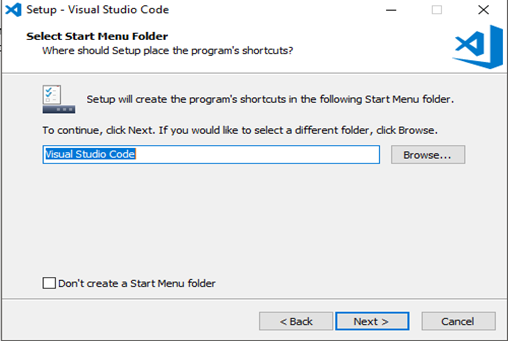


Figure 9

Creating an start menu folder for visual studio code

f) In Select Additional Tasks Dialog Box, check all the boxes and click on Next to continue.

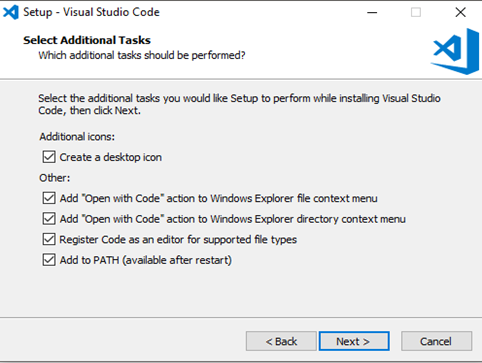


Figure 10

Selection of the additional tasks that was provided

g) Click next to install

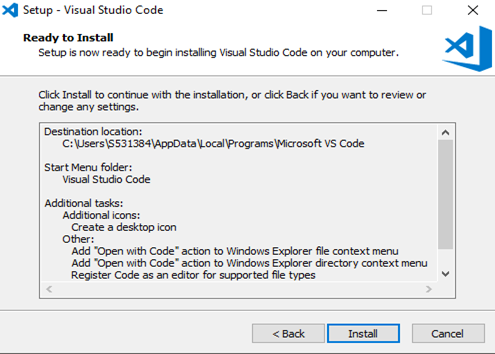


Figure 11

Final step for installation

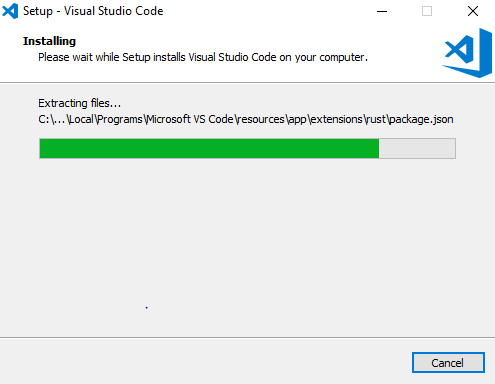


Figure 12

Process of installation

h) After successful Installation, Click Finish

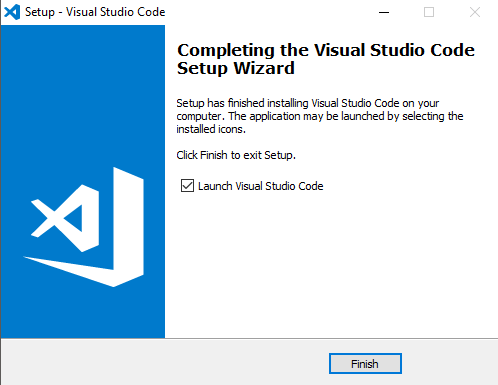


Figure 13

Successful installation of the visual studio code

**2)** **Postman**

a) Download and install postman from the below link

<https://www.getpostman.com/downloads/>



Figure 14

Site for the downloading of the postman



Figure 15

Download option for the postman

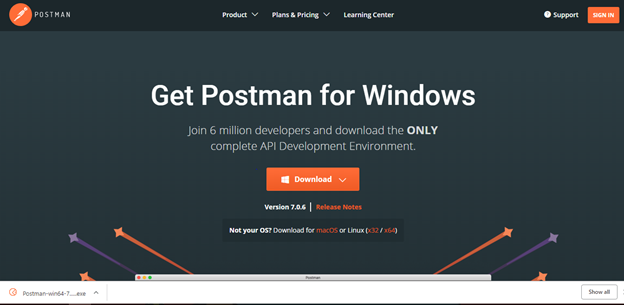


Figure 16

Successful download of the postman

b) After downloading open the .exe file, Postman will begin setting up its environment and finally ready to use.

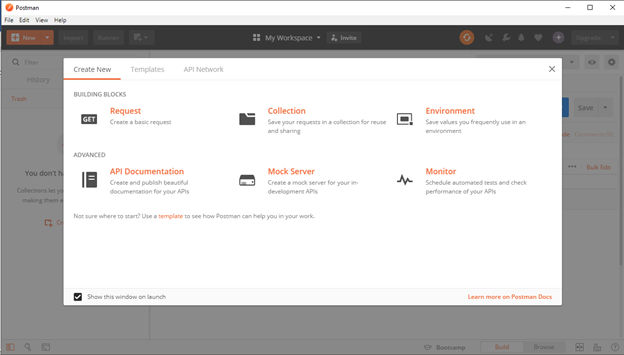


Figure 17

Starting page of the postman after installing the postman

**Tortoise Git**

1. Download and install tortoisegit from the below link

<https://tortoisegit.org/>.



Figure 18

Site for the downloading of the tortoise git

b) After clicking on download, Select the download version for 64 bit windows.

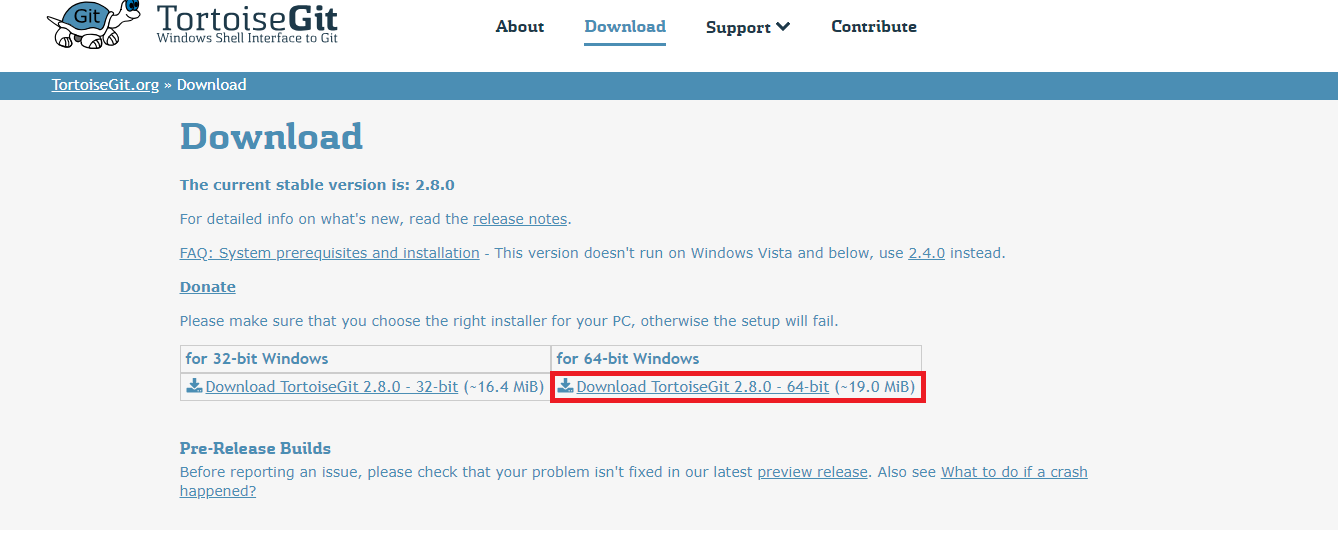


Figure 19

Version to be selected for downloading the tortoisegit

c) After downloading, a pop-up showing the setup of the tool will be appeared on the screen.



Figure 20

Setup of the tortoise git

d) The following screenshots show the steps to install tortoisegit.

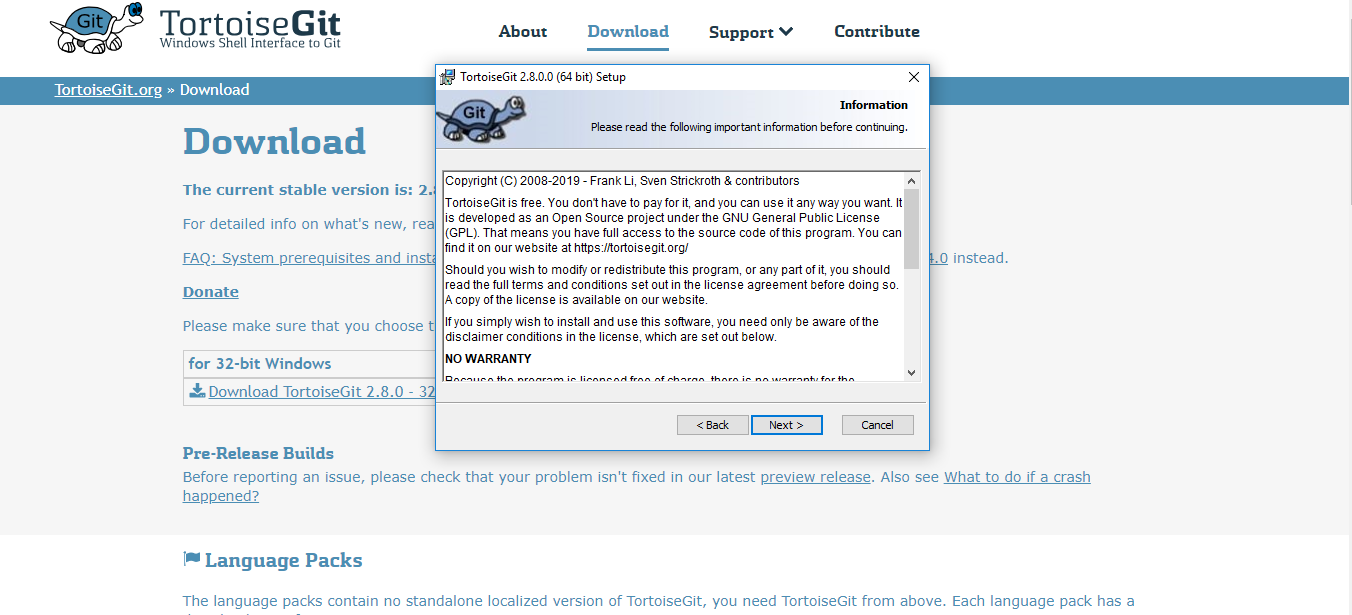


Figure 21

Installation for the tortoisegit

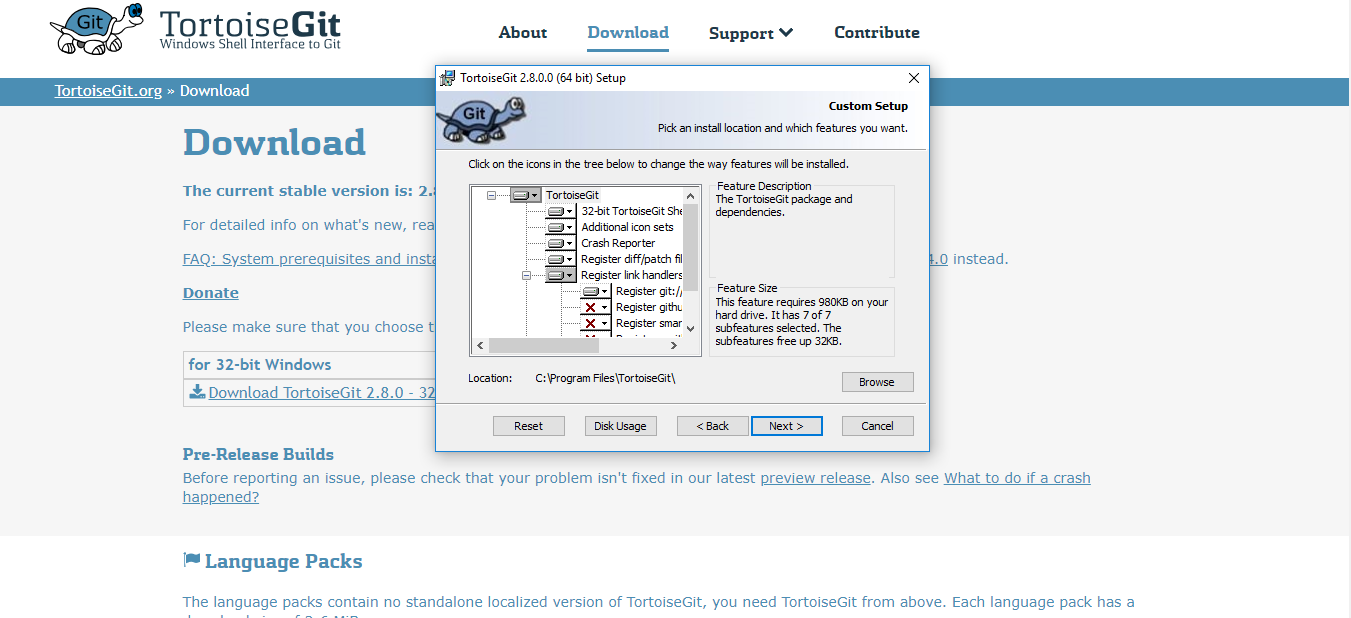


Figure 22

Continuation process for the tortoisegit

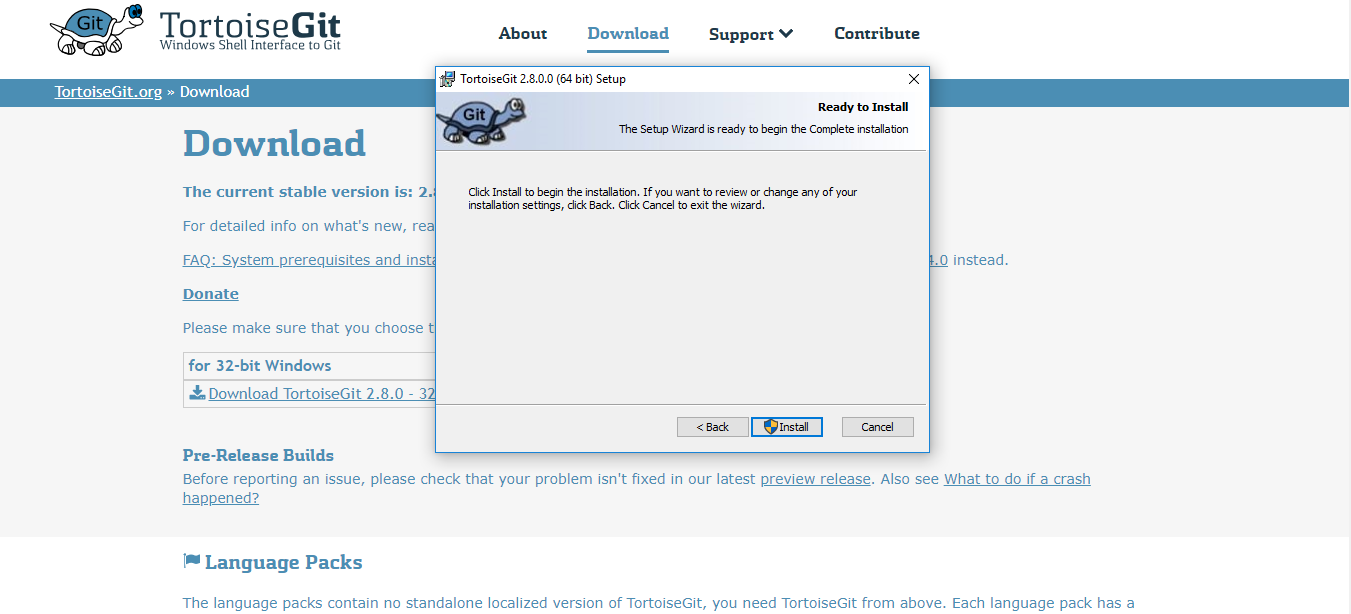


Figure 23

Final step for the installation of the tortoisegit

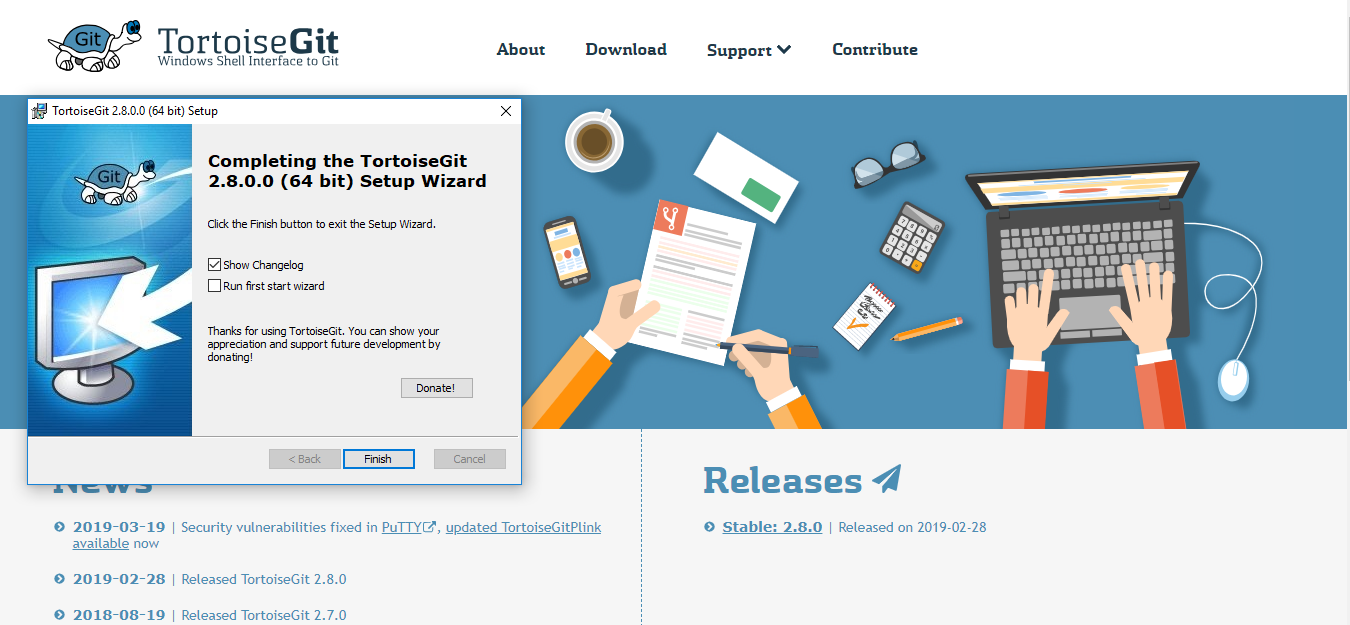
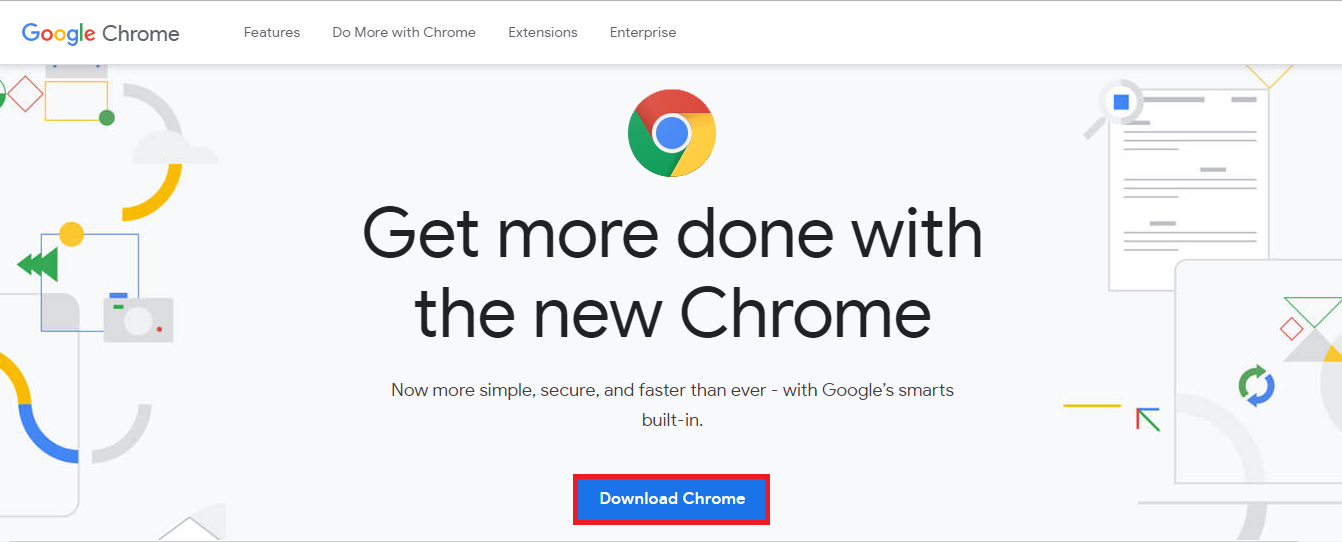
e) Click on ‘Finish’ After clicking on install that the tool is ready to be use

Figure 24

Tortoisegit is ready to use

* **Google Chrome Browser**

1. Download and install Google Chrome from the below link <https://www.google.com/chrome/>.



.

Figure 25

Site to download the google chrome

b) Agree the terms and conditions an install chrome, after clicking on download button.

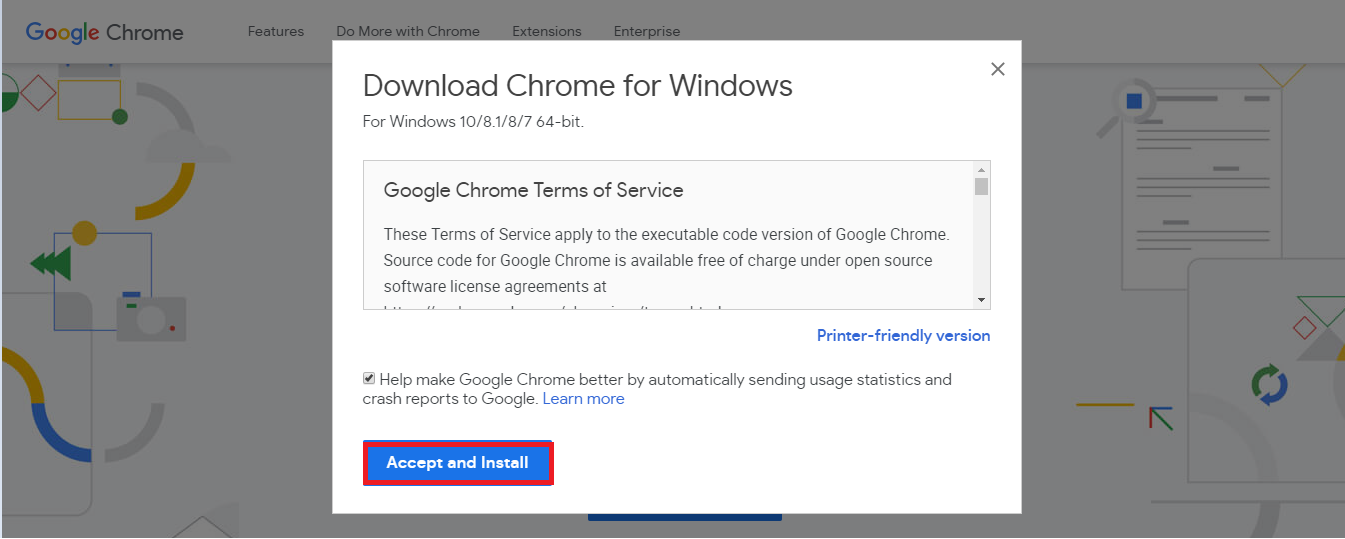


Figure 26

Conditions to be agreed for the google chrome installation

c) After the download of the exe file is done, open the exe file.

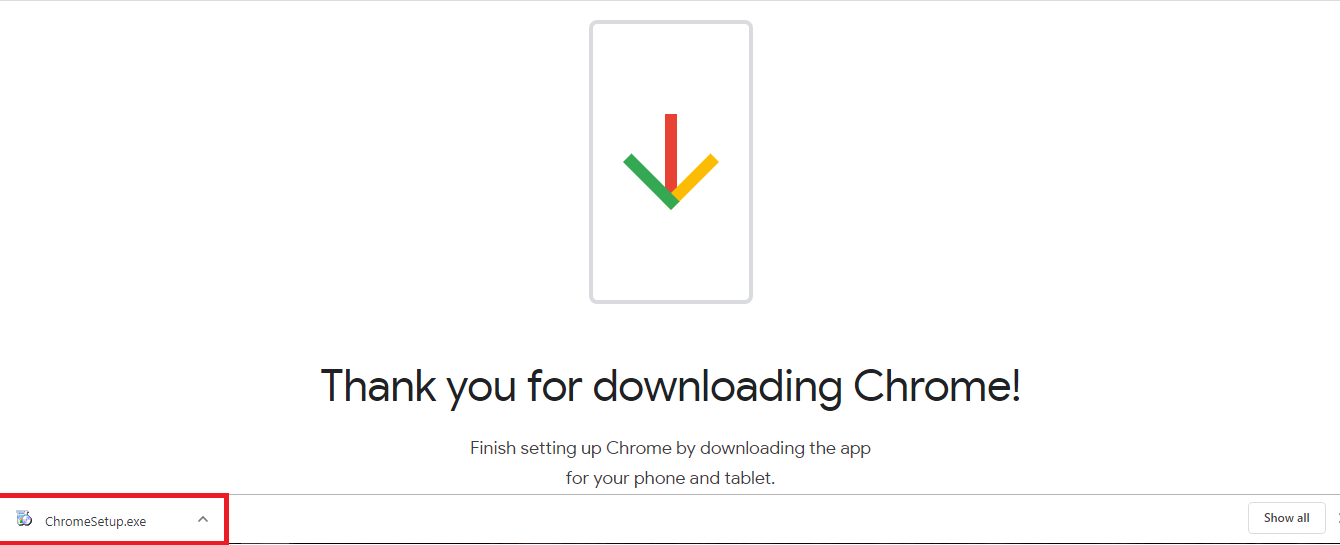


Figure 27

Successful downloading of google chrome

d) The application starts downloading.

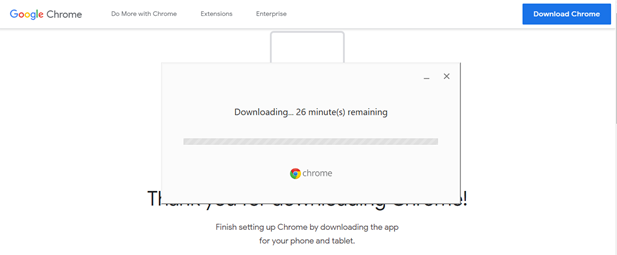


Figure 28

Process of downloading of google chrome

e) After the application is automatically installed a message is displayed.

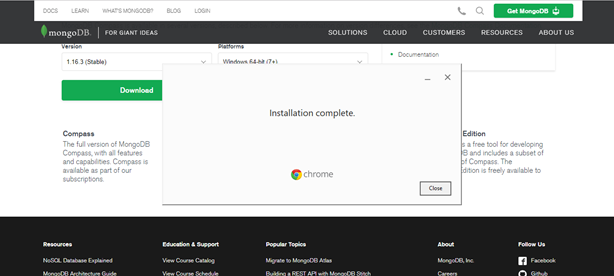


Figure 29

Conformation for successful installation of google chrome.

**MongoDB Compass**

1. Download and install MongoDB Compass from the below link <https://www.mongodb.com/products/compass>.
2. Click on try it now on home page of the Mongo Compass.

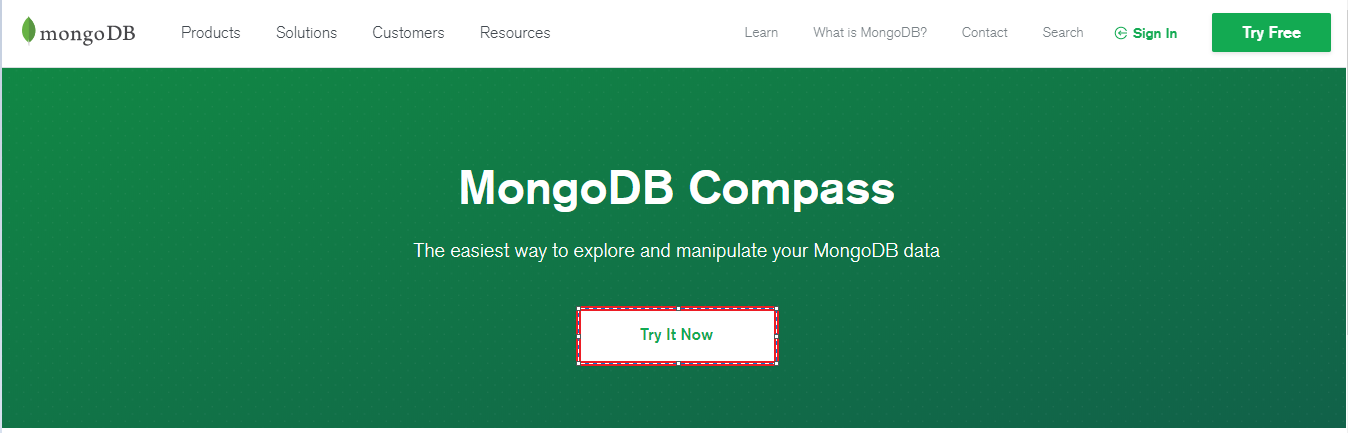


Figure 30

Site for the download and install of MongoDB

c) select the version of the MongoDB compass .

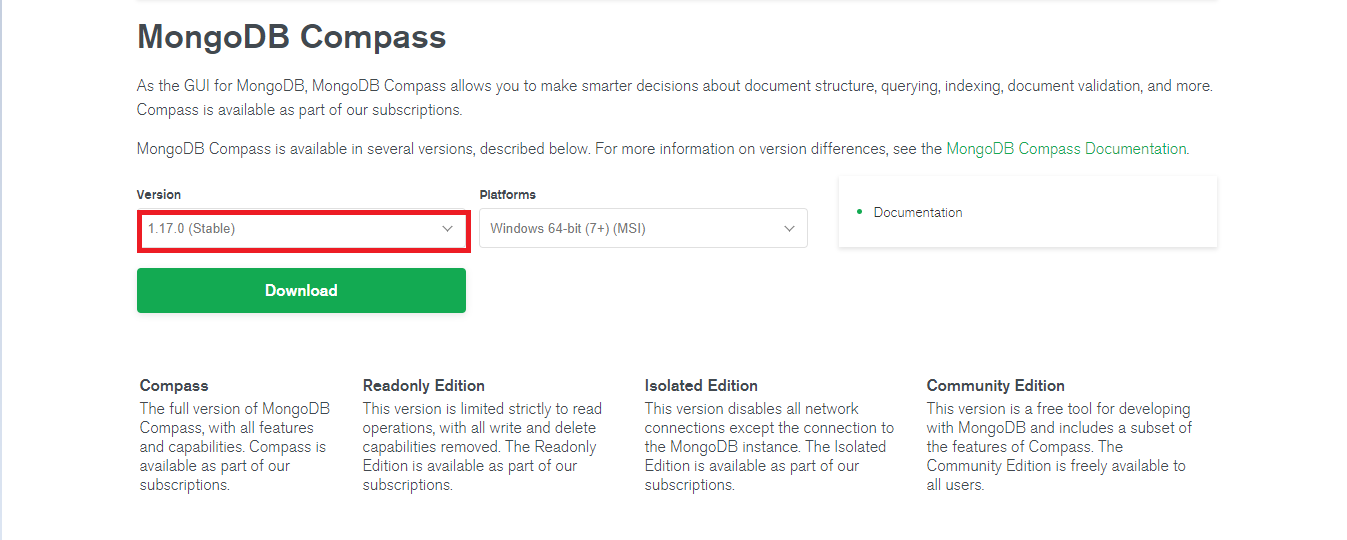


Figure 30

Version Selection of the MongoDB

d) After clicking on the download button, it asks for the details of the user. Fill in the

details and submit the form.

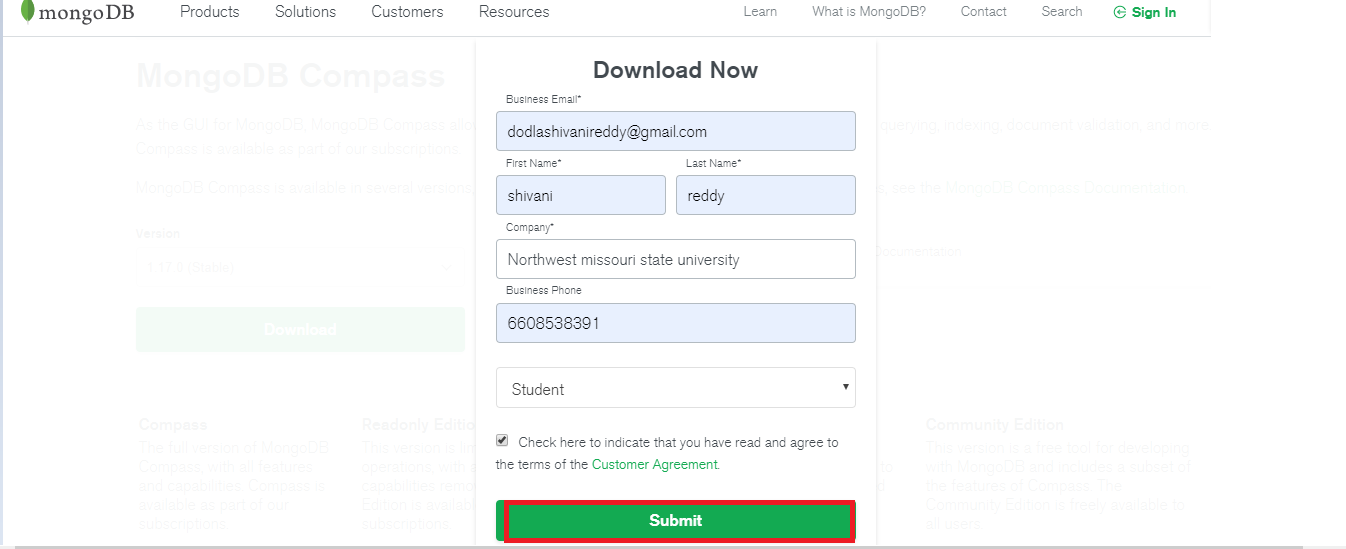


Figure 31

Entering of the user details in the mongoDB

e) After clicking on submit button,The exe file is downloaded.

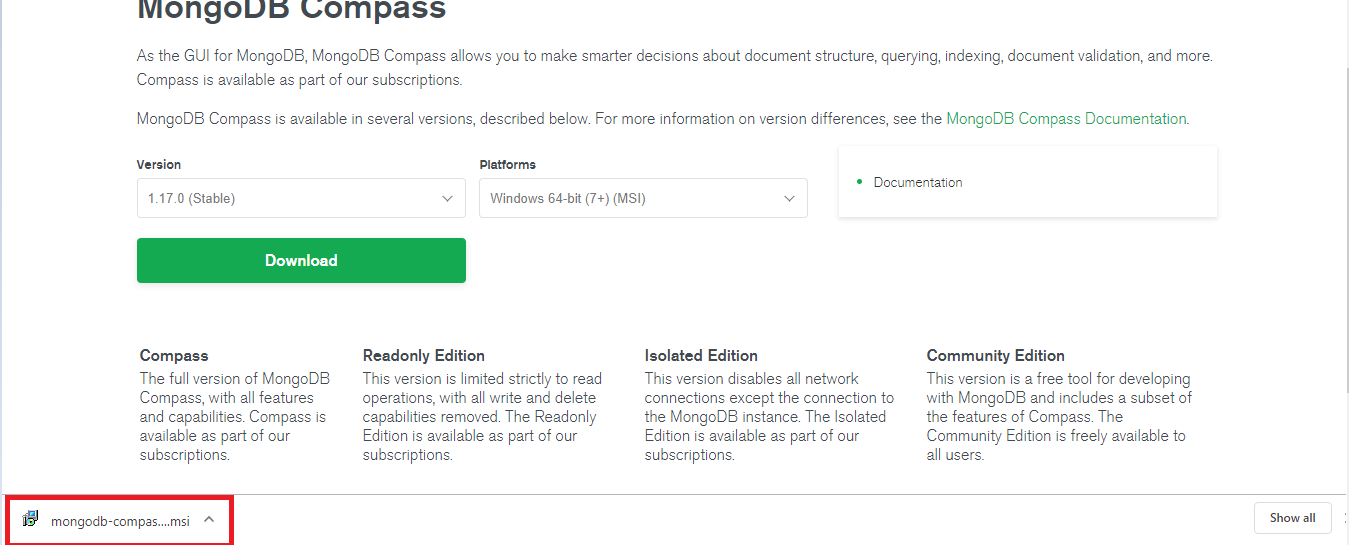


Figure 32

Downloading of the MongoDB

f) The following screen will appear, after clicking on the exe file of Mongodb Compass.

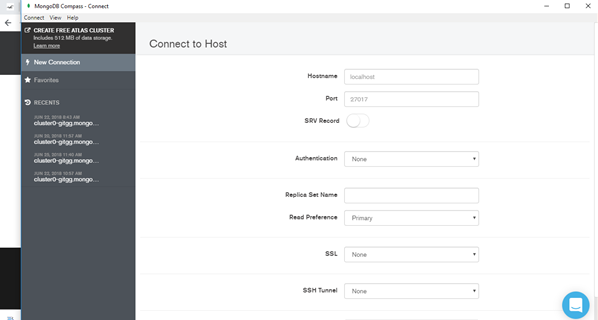


Figure 33

Exe file of the MongoDB compass

**Technologies:**

* **MongoDB Atlas**

a. Register and create an account at<https://www.mongodb.com/>.

b. We will use this cloud service to host our application database.

* **Vue.js**

a. This is a JavaScript framework used to build Single Page Applications.

b. We will use this framework to build our frontend webpages

* **Node.js**

a. This is a JavaScript Platform where we can build web services.

b. We will use this technology to build our backend web services.

* **Heroku**

a. This is a free cloud hosting platform where we can deploy our application.

b. We will be deploying our backend services application to Heroku.

**1.** **Developers’ Manual:**

**1.1 Instructions to execute the project:**

1. Clone the project onto your machine from the GitHub Repository using the following link:

<https://github.com/chvnaveenkumar/Codeword-GDPII.git>

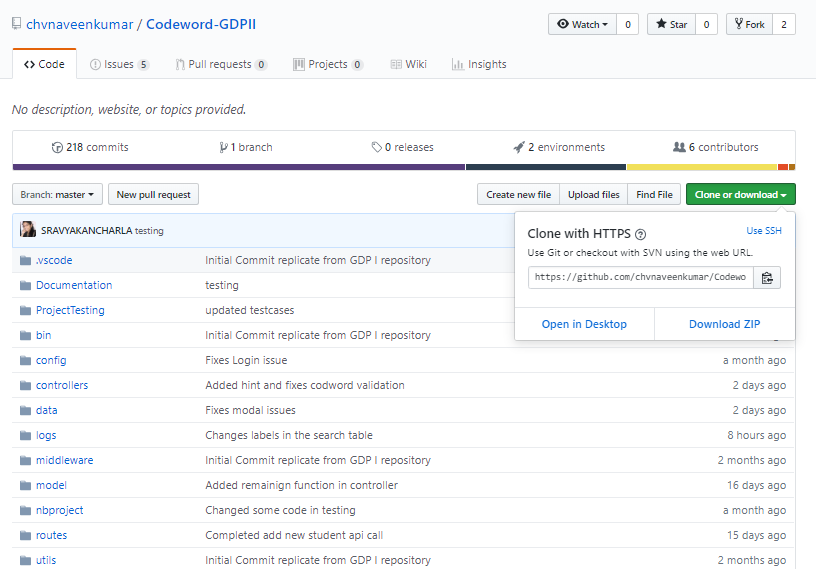


Figure 34

Codeword Git Repository

2. After cloning the project on your machine, you will notice the project folder as follows:

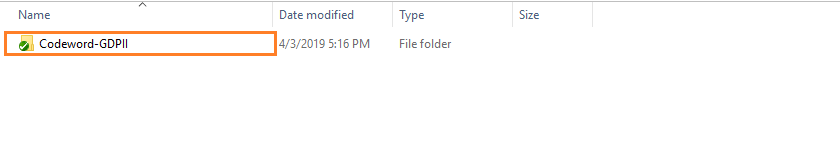


Figure 35

Codeword File after cloning

3. In the “Codeword-GDPII” folder open the folder contents with Visual Studio code as follows:

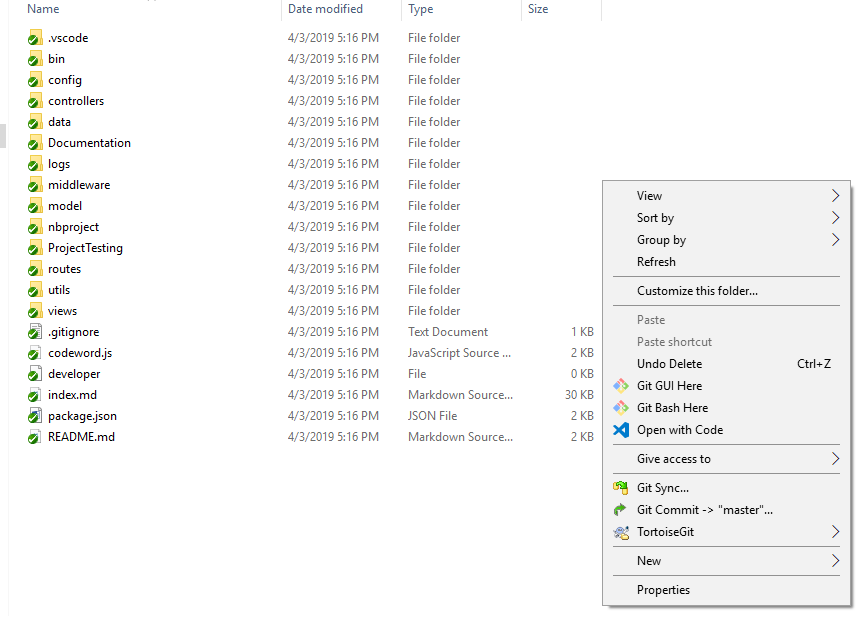


Figure 36

Opening in Visual Studio Code

4. The Visual Studio Code opens the “Codeword-GDPII” as follows:

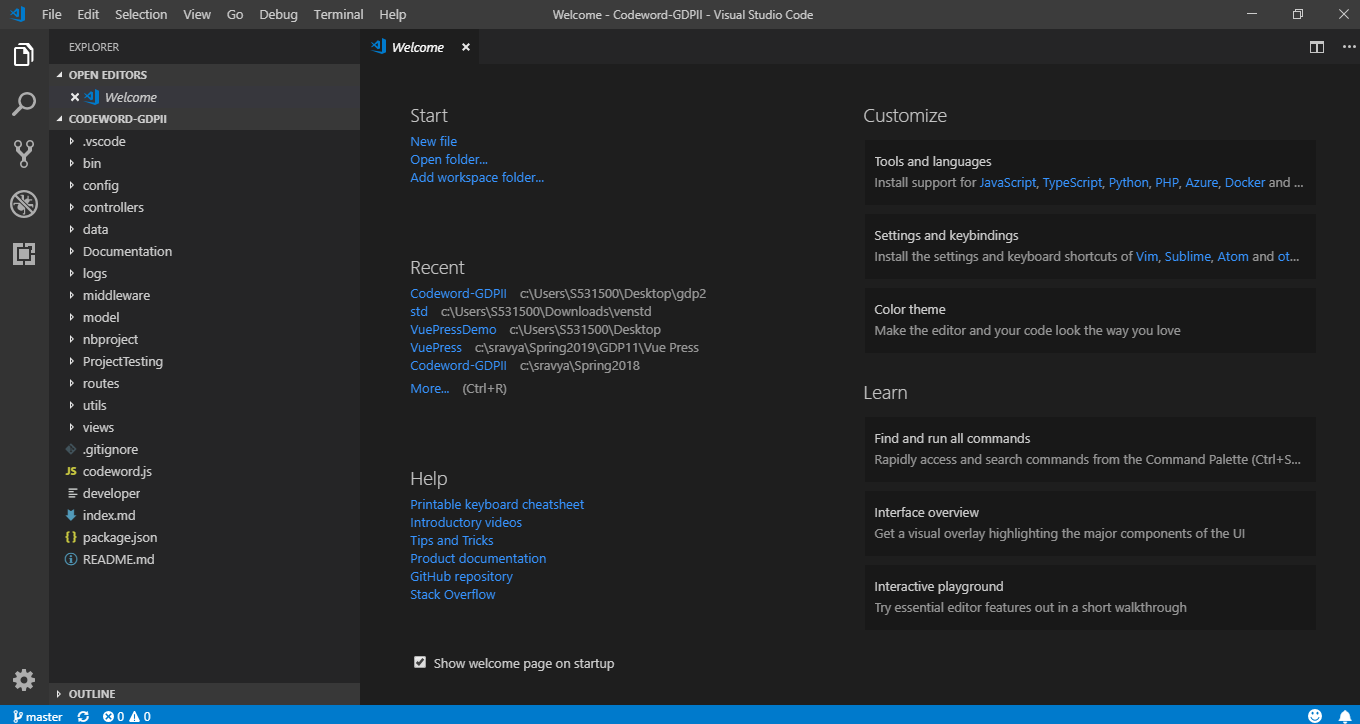


Figure 37

Code in Visual Studio Code

5. To run the project in the Visual Studio Code, click on the “View” option in the navigation bar and click on “Terminal”.

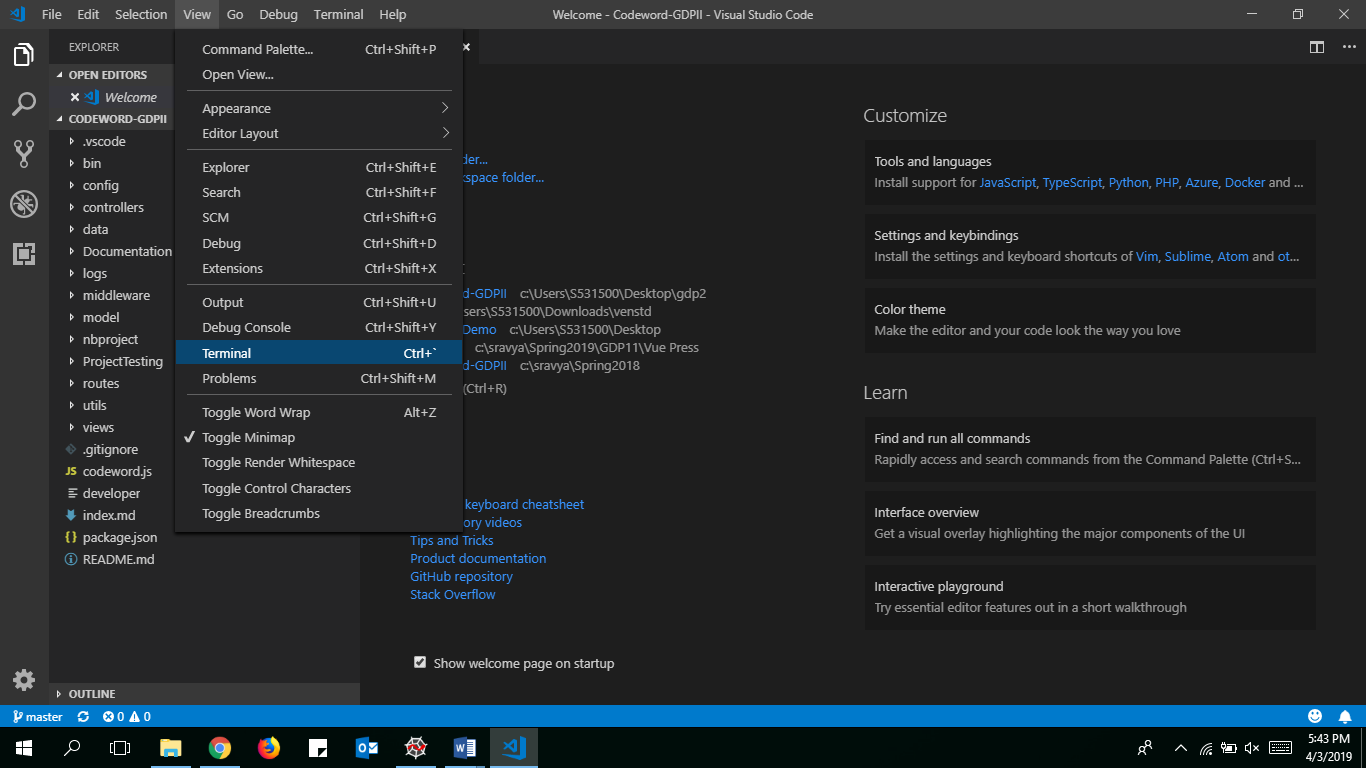


Figure 38

Opening Terminal

6. After clicking on the “Terminal” you will notice a terminal screen at the bottom of the screen in the Visual Studio Code.

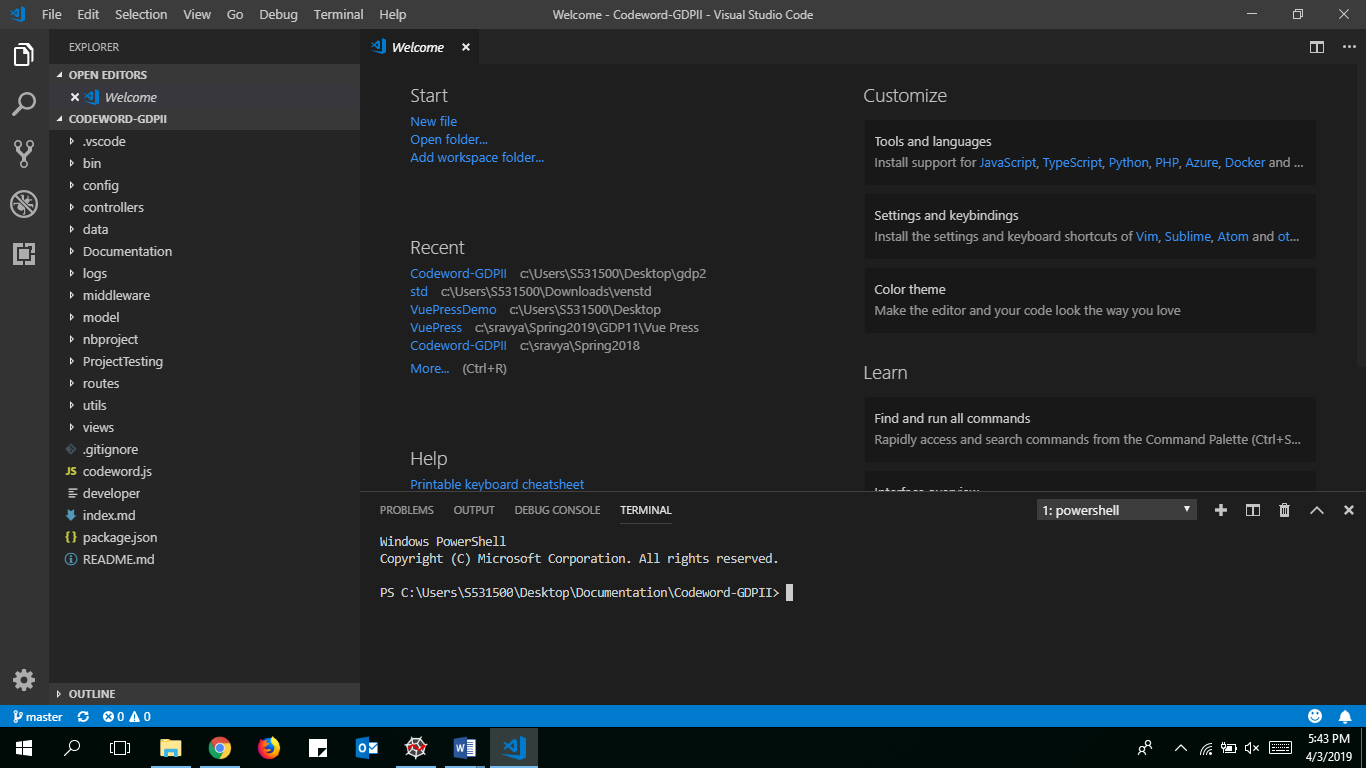


Figure 39

7. In the Terminal window type in **npm install** to install all the necessary dependencies to run the project as follows:

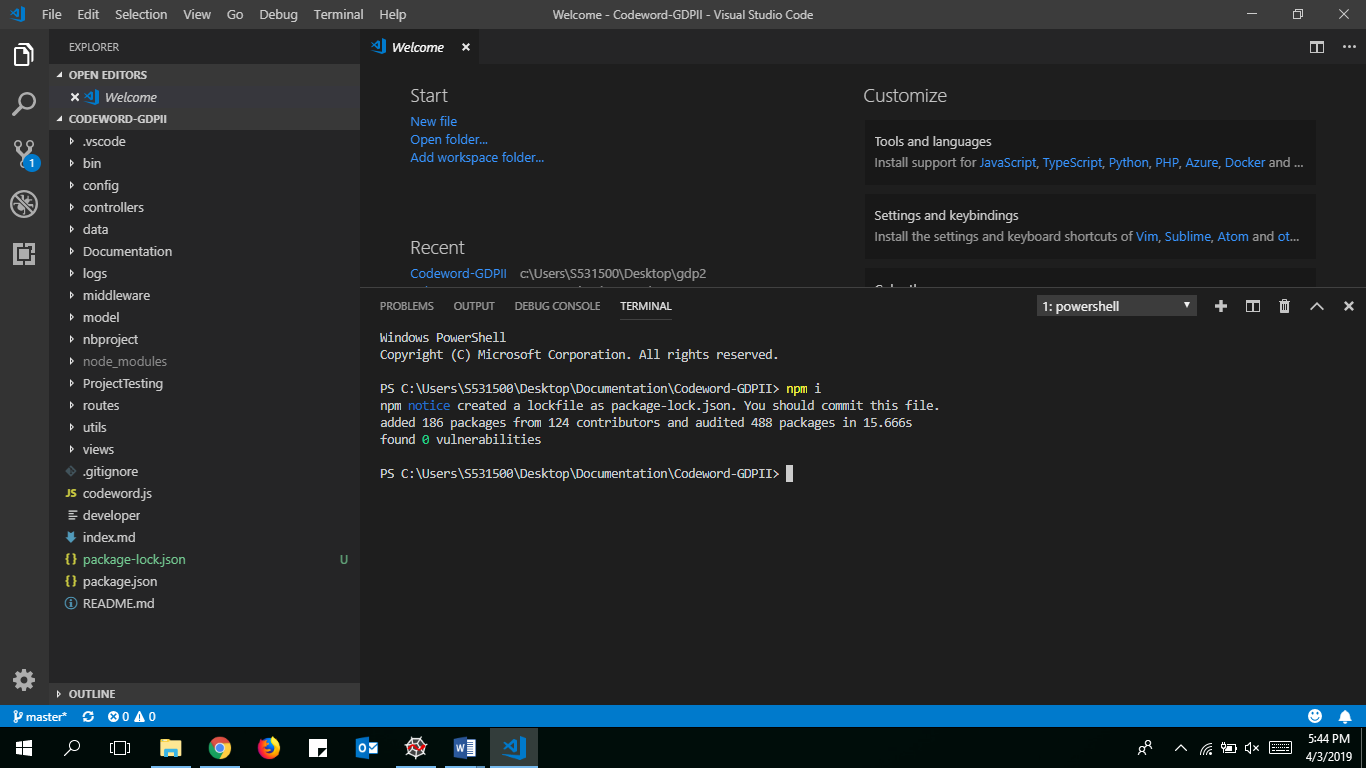


Figure 40

Installing Dependencies

8. In terminal window, type **cd views** to direct path to views folder. Now type **npm install** to install the dependencies to run the project.

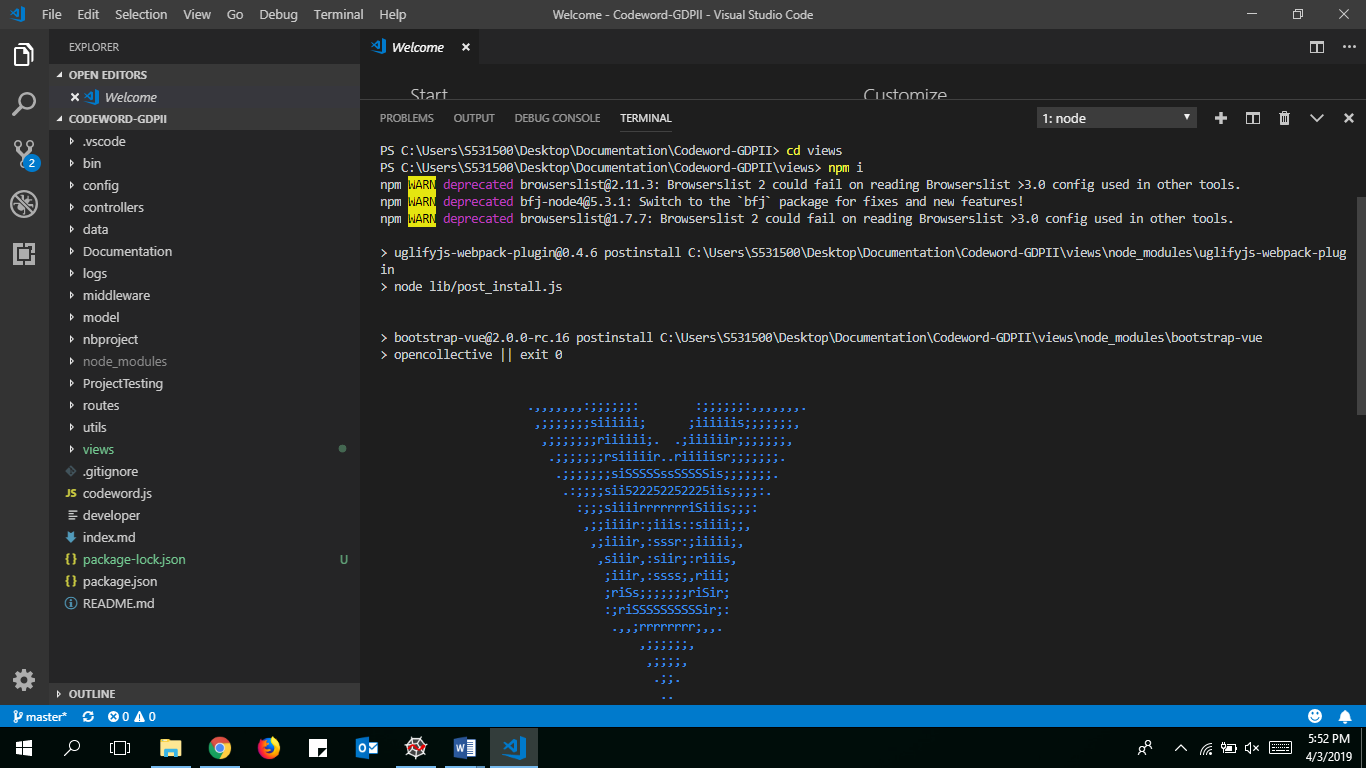


Figure 41



Figure 42

Running Code

9. After installing all the necessary dependencies for the project, type in **npm run dev,**  to connect the project to the backend click f5, as follows:

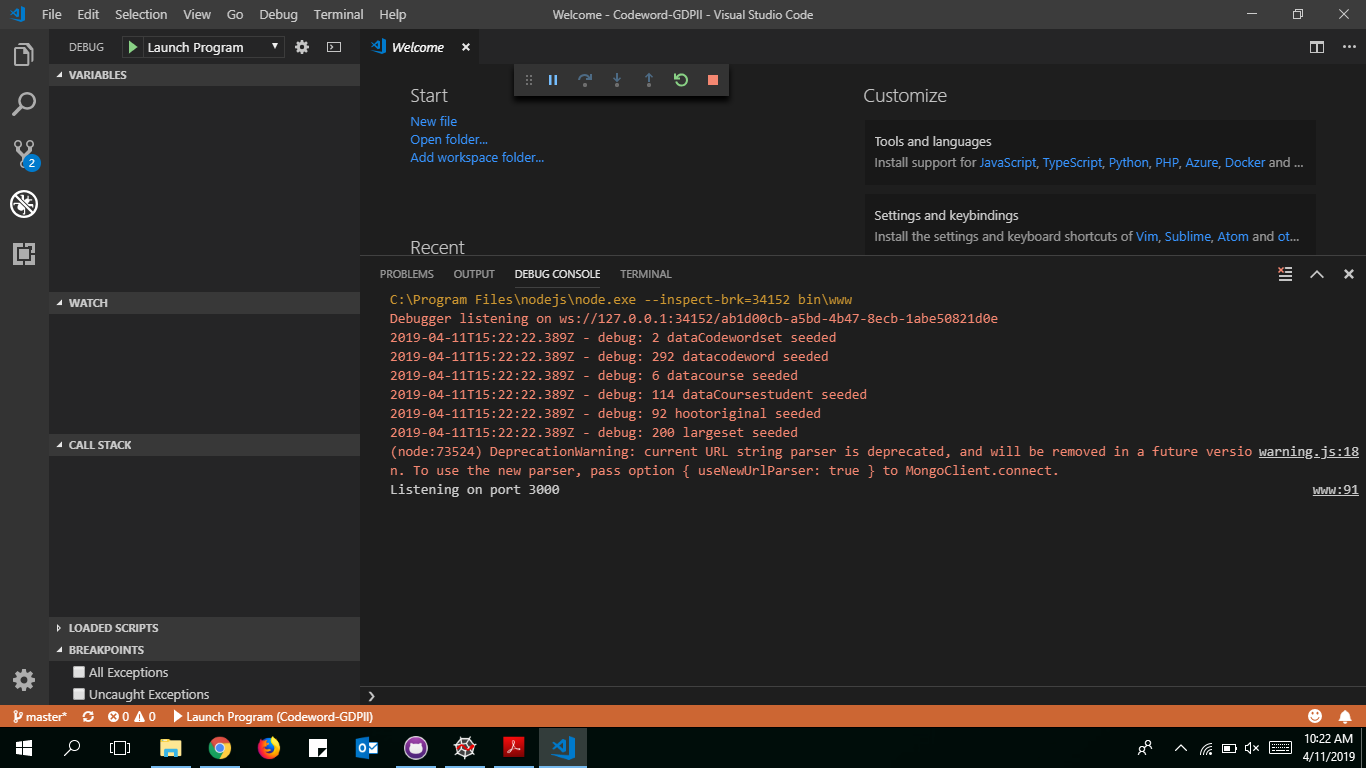


Figure 43

Connecting to Database

10. Now it will redirect to a link in chrome page.

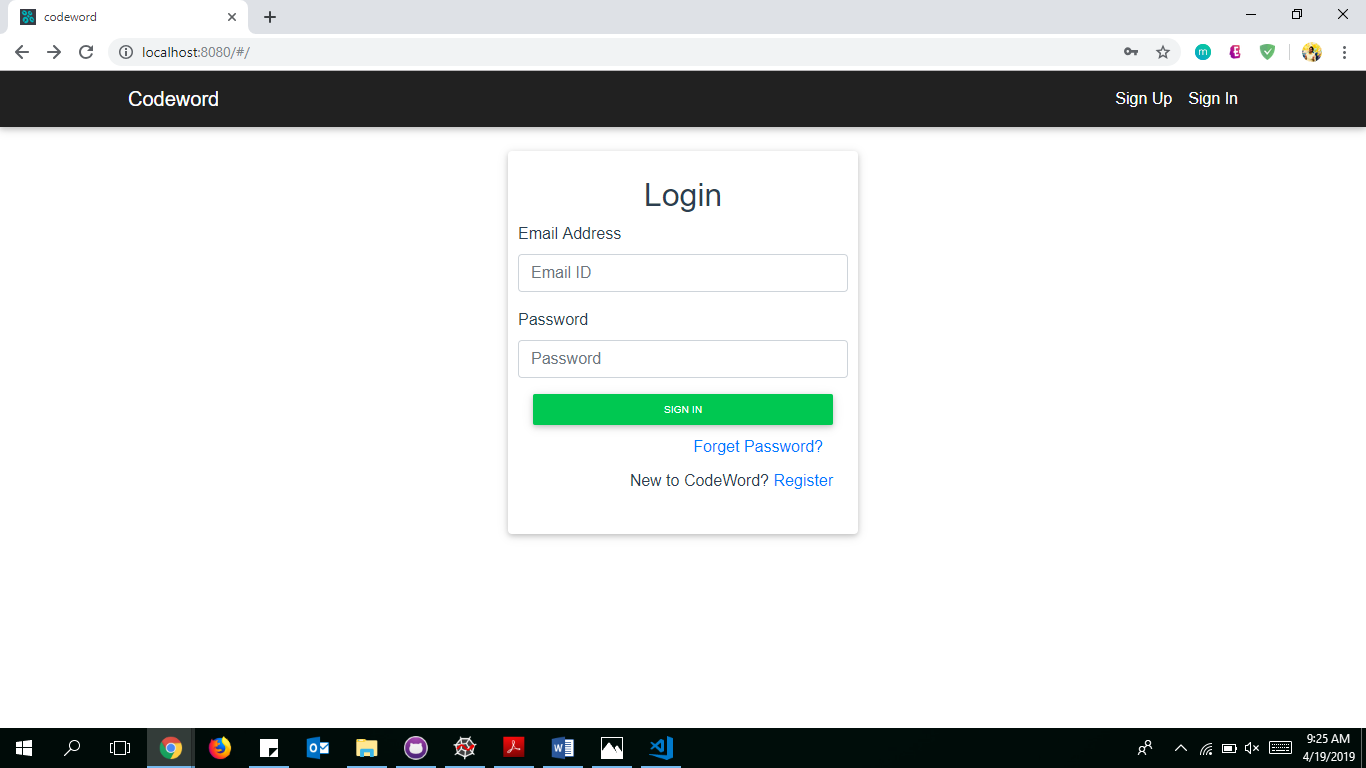


Figure 44

After running code

**1.2 Known Errors:**

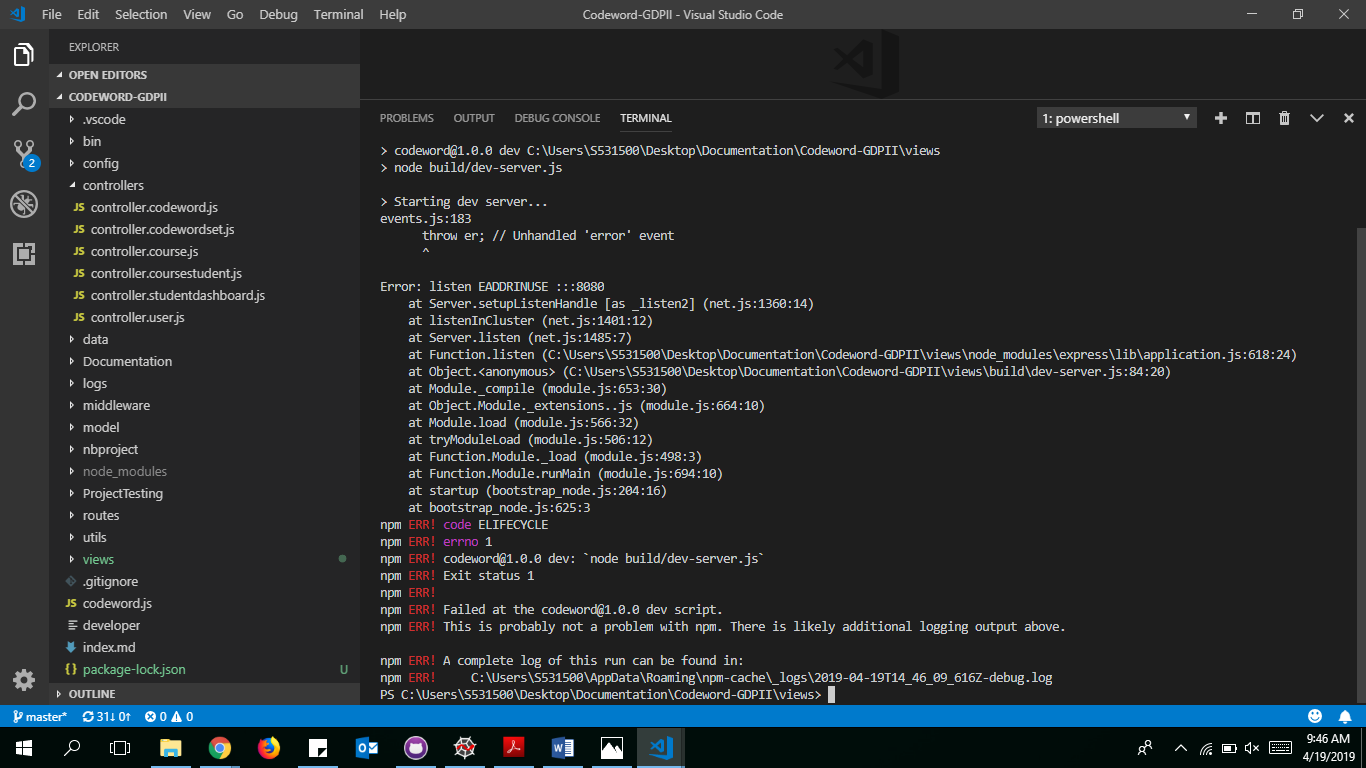
****

Figure 45

Error while running code

If Sometimes you get an error, to stop that error 2 steps need to be followed:

a) netstat -ano | findstr :8080

b) taskkill /PID typeyourPIDhere /F

PID: Process Identification, It will be shown after running the step (a).

Here is the link to refer : <https://stackoverflow.com/questions/39632667/how-to-kill-the-process-currently-using-a-port-on-localhost-in-windows>

**Maintenance Instructions:**

1. After logging into the codeword set, you have to register by clicking on R**egister** button.
2. User can enter .edu mails or gmails.
3. User should enter an email, if user is an instructor then check the instructor check box.
4. User should click on **send temporary password,** so that he will get a temporary password to the email.
5. It will redirect to the login page, where we will get a chance to enter email and temporary password.
6. If user forgets password, he/she can click on **Forget Password** and next page will be entering email and send temporary password button.
7. User should click on **login** button where you will redirect to change password page and will get a chance to change the temporary password.

**Student View:**

1. If user is a student, he/she will find their courses if instructor has already invited them.
2. Once they find their courses, they will be able to click on **Get codeword** button so that they can see their codeword.
3. After getting their codeword they are allowed to take survey.

**Instructor View:**

1. If user is an instructor, he/she will redirect to Instructor Dashboard.
2. In Navbar, user will find buttons as **Instructor Dashboard, Codeword Sets.**
3. If instructor wants to upload a new Codeword set instructor has to click on **Codeword Sets** page.
4. There you will find a default Large and Small codeword sets.
5. If Instructor wants to upload new codeword set then click on **+ UPLOAD CODEWORD SET** button, which pops up a new model box.
6. Instructor can upload a new codeword set by giving a name in given format as below.

**Criteria for Codeword File:**

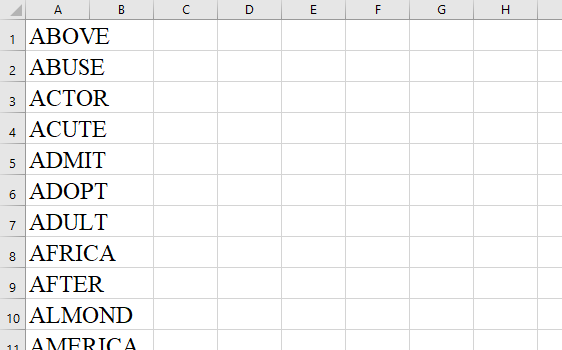
****

Figure 46

Codeword File

* In the Excel sheet codewords should be unique.
* They should be of single word.
* They should be of only characters but not numbers or special characters.
* Excel sheet should not be empty.
* Excel sheet should not have empty cells.

1. Instructor can edit the new added codewordset. He/She can edit, delete or add a new codewords to the set.
2. He/She can search for a codeword in that set or can view with pagination view. It defaults to 10 codewords per page and can change to 25,50 or 100.
3. Codewords can sort by ascending or descending order.
4. For adding a course, In Instructor Dashboard you will have chance to add courses by clicking on **+ ADD COURSE.**
5. Instructor can be able to fill in the details like Course Name, StartDate, EndDate, Uploading student details as per criteria given below, Selecting Codeword list and giving urls are instructors choice.

**Criteria for Creating Course :**

* 1. **Student List File:**

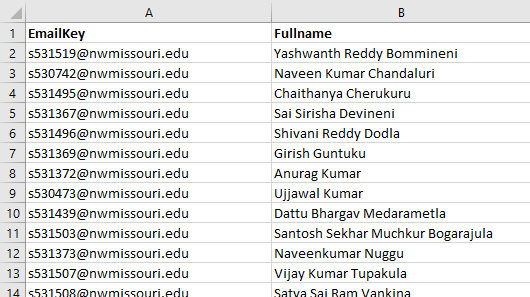
****

Figure 47

Student File Criteria

* It should have Email Key and Student Name in the File.

1. By clicking on **CREATE COURSE** button, it will come back to instructors dashboard page and course will be shown with Course Name, StartDate, EndDate, View and Delete buttons with Acknowledge status.
2. If instructor wants to delete course he/she can click on delete button and will confirm about deleting the course, if clicked on delete then course will be deleted from the view.
3. Instructor can view the course, where he/she can be able to edit course details, able to see the acknowledge status, percentage of students who acknowledged the codeword, search for a student, add, delete or edit student details, and can be able to sort according to email address and student name by clicking on Student Email and Student Name.
4. It defaults to 10 student per page and can change to 25,50 or 100.
5. For adding a new student click on **+ ADD NEW STUDENT** and give student email and student name.
6. In Instructor Dashboard, he/she can be able to see inactive courses by checking **Inactive Courses** button.
7. In NavBar, you will find logout button to logout.

**Test Cases:**

* + - 1. Entering login credentials
      2. Entering email address without password
      3. Clicking on sign in button without entering credentials
      4. Entering wrong email address
      5. Clicking on forget password
      6. Entering wrong email address
      7. Don’t enter email and click send password reset email
      8. Able to see the active courses that instructor teaches.

**Results and analysis of test cases:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Scenario** | **Test Case iD** | **Test Case Description** | **pre-requisites** | **Steps** | **Actual** | **Expected** | **Remarks** |
| Login | 1 | Entering Login Credentials | URL: http://localhost:8080/# | 1. Enter Url In Browser 2. Enter Email Address 3. Enter Password  4. Click on signIn button | Navigate Next Page | Navigate to Next Page | Success |
| Login | 2 | Entering Email Address without password | URL: http://localhost:8080/# | 1. Enter Url In Browser 2. Enter Email Address 3. Click on signIn button | Should show error message as Invalid user | Invalid user | Success |
| Login | 3 | Clicking on signin button without entering credentials | URL: http://localhost:8080/# | 1. Enter Url In Browser 2. Click on signIn button | Should show error message as Invalid user | Invalid user | Success |
| Login | 4 | Entering wrong Email Address | URL: http://localhost:8080/# | 1. Enter Url In Browser 2. Enter wrong Email Address 3. Enter Password  4. Click on signIn button | Should show error message as Invalid user | Invalid user | Success |
| Forget Password | 1 | Clicking on forget password | URL: http://localhost:8080/#/forgetpassword | 1. Enter Url in Browser  2. Enter Email Address 3. Click on Send Password Reset email | Show Sent temporary password to email.  Redirect to Login page. | Show Sent temporary password to email.  Redirect to Login page. | Success |
| Forget Password | 2 | Entering wrong Email Address | URL: http://localhost:8080/#/forgetpassword | 1. Enter Url in Browser  2. Enter Email Address 3. Click on Send Password Reset email | Should show error message as Invalid user | Should show error message as Invalid user | Success |
| Forget Password | 3 | Don’t enter email and click Send password reset email | URL: http://localhost:8080/#/forgetpassword | 1. Enter Url in Browser  2. Click on Send Password Reset email | Should show error message as Invalid user | Should show error message as Invalid user | Success |
| Instructor Dashboard | 1 | Able to see the active courses that intructor teaches. | Should be logged in as Instructor |  |  |  | Success |

**Deployment:**

Before you can deploy your app to Heroku, you need to initialize a local Git repository and commit your application code into github repository. For our project, we have created a Git repository where you can follow the steps below to deploy app to Heroku.

1. Using the following link, clone the project onto your machine from the GitHub Repository.

<https://github.com/chvnaveenkumar/Codeword-GDPII>

2. After cloning the project on your machine, you will notice the project folder as “ Codeword-GDP II”.

3. Login to the Heroku online by going into the link “<https://id.heroku.com/login>“.

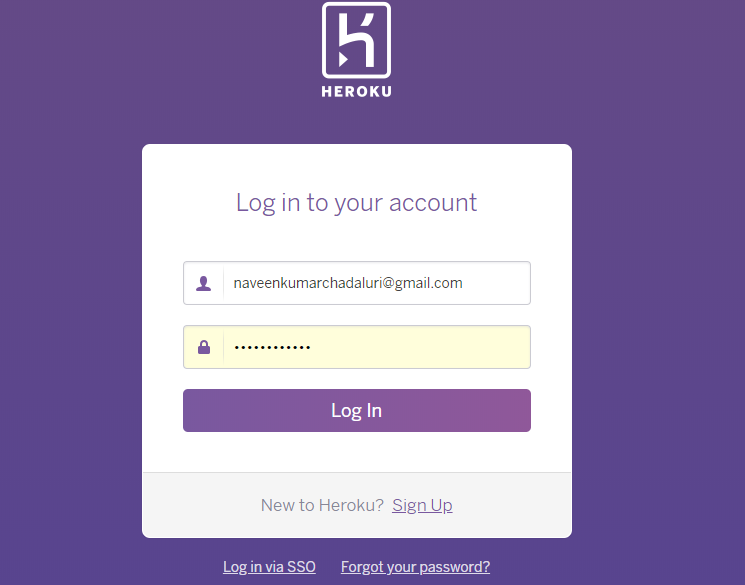
4. Enter your credentials as shown in the image below:

Figure 48

Entering Credentials for heroku

5. Once you are logged in, click “ New” button on right top corner and then select “ Create new app” .

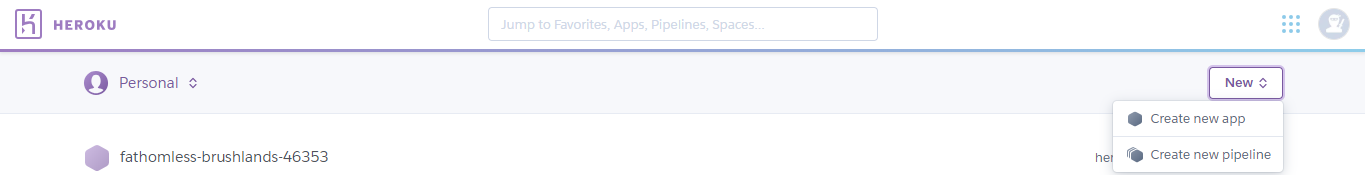


Figure 49

Creation of new app in the heroku

6. Enter the app name “codewordappnew” or any name with and then click on “ Create app”. If app name “You can choose any region (It is not a mandatory).

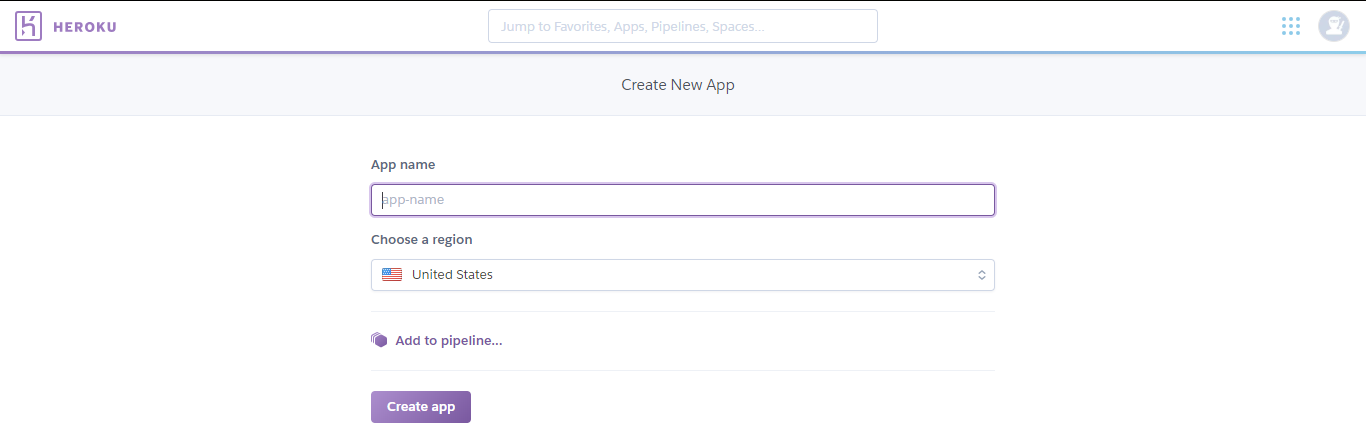


Figure 50

Creation of app by using the name and region

7. Once the app is created into the Heroku, you can easily add a remote to your local repository with the heroku git:remote command.



Figure 51

Adding of remote to the local repository

8. To deploy your app to Heroku, you typically use the git push command to push the code from your local repository master branch to your heroku remote, like so:



Figure 52

Push command to push the code

9. Once the app is pushed to heroku, click on “ Open app” in the Heroku page as shown below:

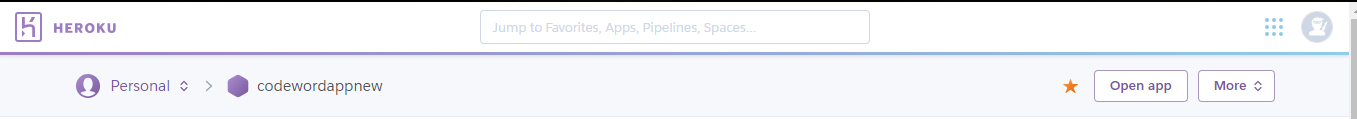


Figure 53

Open app to view the app in the heroku

b) Deploy connecting to GitHub using heroku:

1. Using the following link, clone the project onto your machine from the GitHub Repository.

<https://github.com/chvnaveenkumar/Codeword-GDPII>

2. After cloning the project on your machine, you will notice the project folder as “ Codeword-GDP II”.

3. Login to the Heroku online by going into the link “<https://id.heroku.com/login>“.

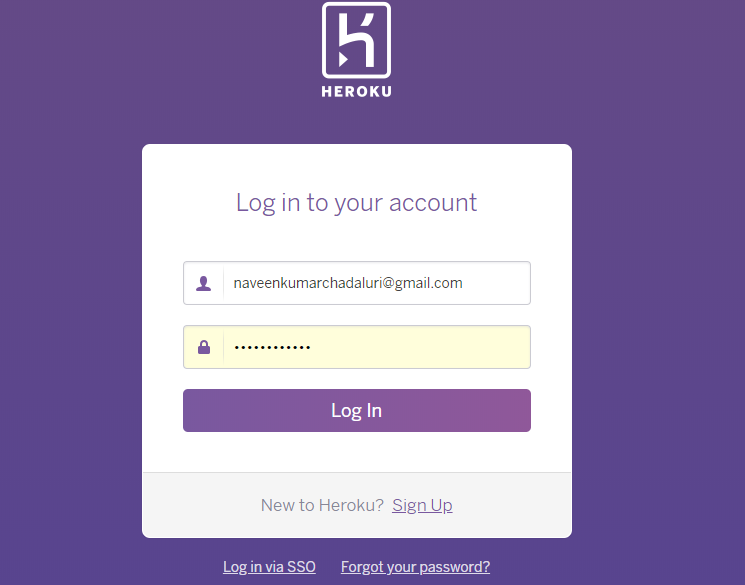
4. Enter your credentials as shown in the image below:

Figure 54

Entering of credentials for heroku

5. Once you are logged in, click “New” button on right top corner and then select “ Create new app” .

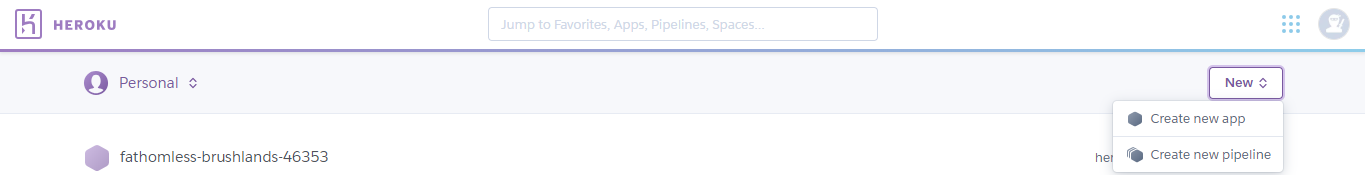


Figure 55

Creation of new app in the heroku

6. Enter the app name “codewordappnew” or any name with and then click on “Create app”. If app name “You can choose any region (It is not a mandatory).

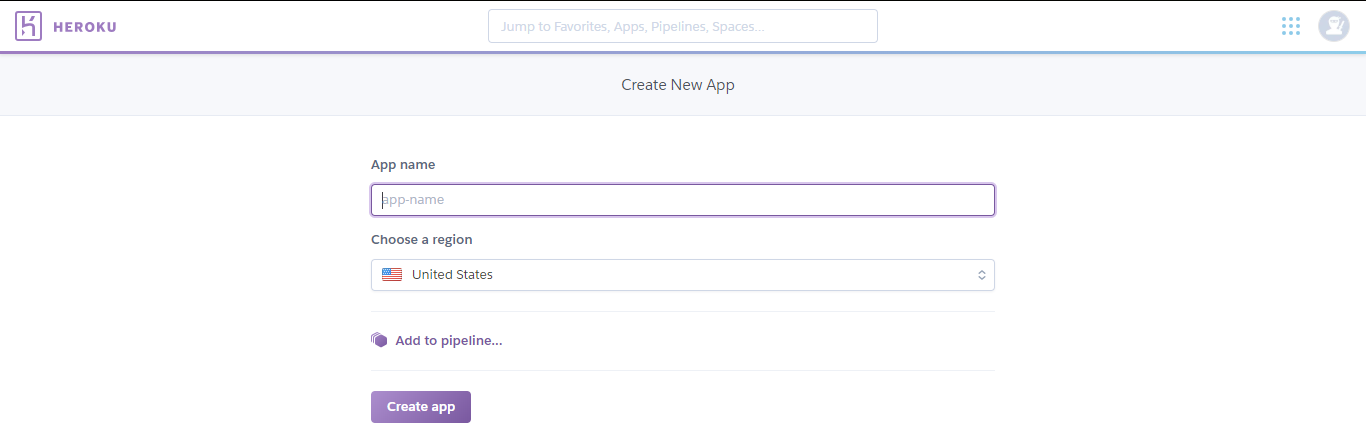


Figure 56

Choosing of app name and region for the new app

7. Select the Deploy option in the menu bar. Select the deployment method “GitHub - Connect to GitHub”.

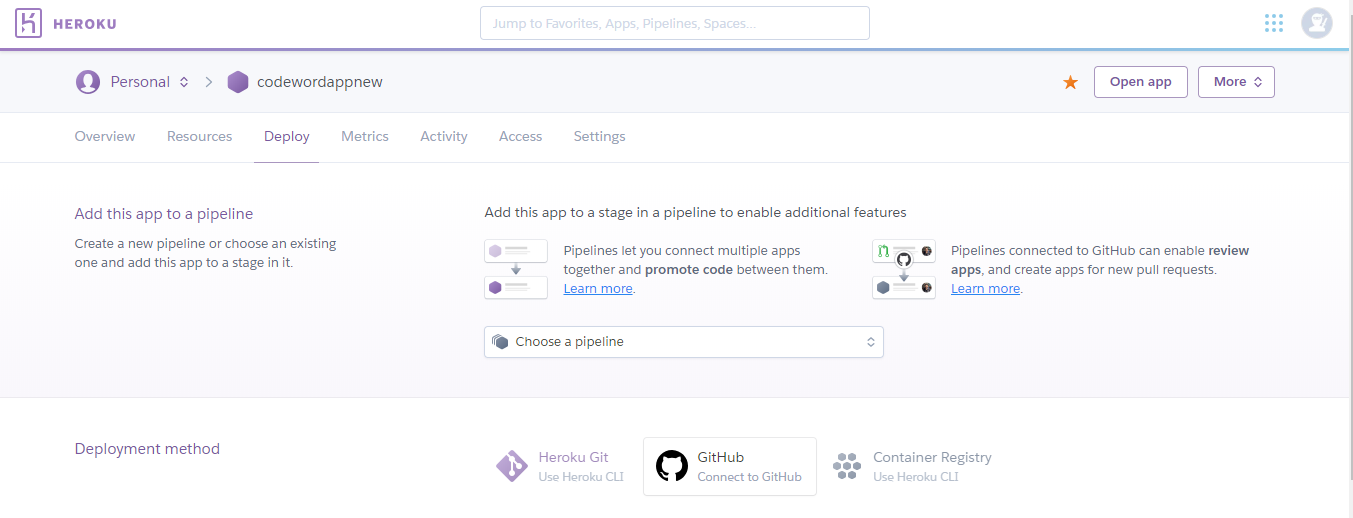


Figure 57

Deployment method of github

8. Enter the repo-name then search the project “Codeword II” and click on “Connect” to connect your app to GitHub from Heroku.

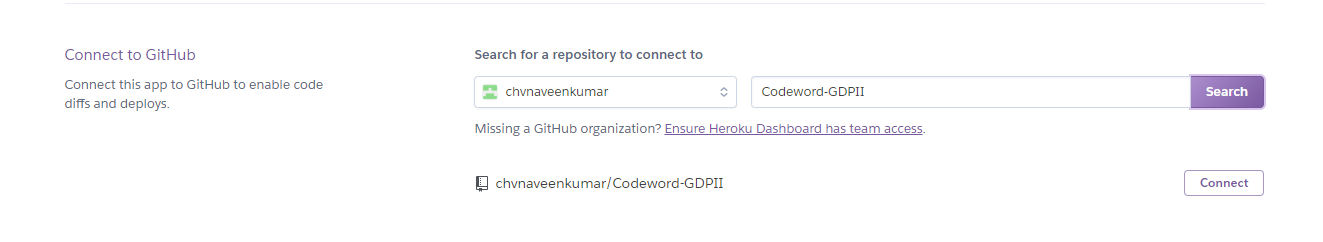


Figure 58

Connection of app to the GitHub from the heroku

Deployment URL: <https://gdpcodeword.herokuapp.com/#/>