WEEK – 1

1. **Digital Transformation Success Story of Netflix**

- Netflix is an American subscription video on-demand over-the-top streaming service owned and operated by Netflix Incorporated company. The service primarily distributes films and television series produced by the media company of same name from various genres.
- It is available internationally in multiple languages. Netflix was launched on January 16, 2007 as a Streaming media company. To enjoy the Netflix services, customers need to subscribe [illegible] and can experience the services provided.
- Its headquarters are situated at Los Gatos, California, U.S., operating in a geographic expansion [illegible] and nearly 238.39 million active users are using the service as of June 30, 2023 reports.
- Netflix's founders were Reed Hastings and Marc Randolph. Current CEO of the Netflix company is Greg Peters. Netflix was once a DVD rental service, Now Netflix has transformed into a leading global streaming platform.
- By leveraging investments with digital technology and data analytics, Netflix personalized its content recommendations and optimized its streaming quality.
- This digital technology also enhanced in creating the original content based on viewer preferences. This led to the exponential growth and dominance in the video streaming industry.

Digital Transformation: Digital transformation is a strategic approach that aims to improve operational efficiency, enhance decision-making, and create new business opportunities by integrating traditionally separate IT and OT systems.

Eg: [illegible] attendance management system.

Information Technology: IT refers to the use of computers, networks, and other physical devices, infrastructure and processes to create, process, store, secure and exchange all forms of electronic data. The commercial use of IT encompasses both computer technology and telecommunications.

Operational Technology: It is the practice of using hardware and software to control industrial equipment and it primarily interacts with the physical world.

Eg: Manufacturing, transportation and more.

IT/OT Convergence:

- It is the integration of information technology systems with Operational technology systems.
- IT systems are used for data centric computing while OT systems monitor events, processes, devices and make adjustments in enterprise and industrial operations.
- IT/OT Convergence involves transferring of information across both IT and OT networks.
- This connectivity can be used to enhance the value of the resource in addition and progress an organization further towards our business goal.

3. Case Study of Digital transformation through IT and OT convergence:

IDEA

A large Automotive parts manufacturing company in industry decided to undergo a digital transformation by integrating its Information technology (IT) systems & Operational technology (OT) to improve:

- Productivity
- Safety &
- Overall efficiency

Digital transformation challenges faced by this company:

- Continuous evolution of customer needs: Customer's expectations and demands have evolved so when organization put points of effort into digital transformation, customer needs can change throughout that as they are constantly looking for more intuitive and enhanced services.
- Budget constraints: Another challenge of digital transformation is the high cost that comes with it. As this is a huge investment, organizations need to carefully plan the budget and come up with a strategy that will enhance and respond to customers and organization's needs.
- Lack of manpower: When an organization starts digital transformation, lack of manpower is required to perform different tasks at various levels. Considering how complex digital transformation strategies are, the right skill and knowledge set are required to implement the necessary changes.
- Security concerns: As organizations adopt newer tools, digital processes and cloud based technology, they are exposed to higher levels of risk. Consequently, they are required to implement higher security measures and improve their cyber security to defend themselves against threats. For protecting data and other valuable assets of an organization [illegible] and regulatory compliance.
- Inflation and supply chain issues: Recent inflation has become more complex during [illegible]. If increased costs are to be passed on to the buyers, then demand typically falls, therefore fewer goods and services can be supplied by producers. This decreases the inflows and outflows of the industry.

Digital transformation implementation to overcome the above difficulties:

- **Meeting the customers' needs** – Organizations should keep up to date with the current trends and design the new suggestions by satisfying their needs.
- **Digital transformation budget planning** – [illegible] financial resources to facilitate the processes involved. A budget should be prepared based on overall goals and needs, strategies, priorities, timelines, planned outcomes and returns of investments etc.
- **Creating job opportunities** – [illegible] employees [illegible] company. By providing employees opportunities [illegible] the requirement. This can overcome the matter of unemployment.
- **Digital transformation security** – Organizations should implement security controls and policies, invest in new tools and technologies, implement risk management and train employees.
- **Changing the way decisions are made** – Data and analytics have played a crucial role in IKEA's digital transformation, as they have become increasingly embedded in the company's decision-making processes. By leveraging data, the company has been able to modernize its inventory management, logistics, fulfillment, and overall supply chain operations. This has led to the adoption of new working practices and improvement [illegible].
- **Embracing AR and VR technologies** – IKEA has been experimenting with augmented reality (AR) and virtual reality (VR) technologies to enhance customer experiences. The company launched [illegible] to help customers visualize how furniture fits in their homes and has acquired California-based Geomagical Labs to develop a 3D home visualization tool. The tool aims to democratize home design, making it accessible to everyone through smartphone devices.

Final Outcomes

- **Increased operational stability** – [illegible] regarding production losses.
- **Enhanced productivity** – The digital integration streamlined workflows and minimized the downtime, leading to a 15% increase in overall productivity.
- **Improved quality control** – Improving the quality control by controlling the process regarding defects and enhancing product quality.
- **Predictive maintenance** – After improving the quality of the product, predictive maintenance is very much important to reduce unplanned downtime by [illegible].
- **Skill development** – Proper training has to be given for the [illegible] to adapt to new technologies and systems.

Conclusion

In conclusion, IKEA's digital transformation journey serves as an inspiring example of how legacy retail brands can successfully adapt to changing consumer needs and business environments.

By staying true to its core values while embracing new technologies and practices, IKEA has managed to reinvent itself for the future, paving the way for continued success in the digital age.

Here is the final stage that is conclusion for this is to achieve the target in monetary terms, as well as in terms of good will. By focusing regarding production process by taking proper measures in adopting new technologies and providing more manpower for the required workforce, proper quality control for controlling regarding production issues etc. this leads to the fruitful market.

Hence customer service and satisfaction are the key to have a huge success in the business.

3. Identify the Typical Process of Workflow that can be Automated.

Nintendo

Nintendo Co., Ltd. is a Japanese multinational video game company headquartered in **Kyoto**. It develops, publishes and releases both video games and video game consoles. Nintendo was founded in 1889 as Nintendo Karuta by craftsman Fusajiro Yamauchi. Since then, Nintendo has produced some of the most successful consoles in the video game industry, such as the Game Boy, the Super Nintendo and Switch.

Here are some typical processes that could be automated using various tools particularly in the context of the production industry like Nintendo:

1. Conceptualization and Design:

- Automated Idea Generation: AI-driven algorithms can analyze popular trends, player preferences, and market data to generate game concept ideas.
- Procedural Level Design: Tools can automatically generate game levels using predefined rules and algorithms.

2. Pre-production:

- Asset Generation: AI tools can create textures, models, and other art assets based on design specifications.
- Script Writing: AI can assist in generating dialogues, narratives, and scripts based on the game's story and genre.

3. Production:

- Version Control: Automated version control systems help manage source code, assets, and ensure synchronization among team members.

- **Bug Detection:** AI-powered testing tools can identify bugs, glitches, and performance issues in the game code.
- **Build Automation:** Continuous integration tools can automatically compile, build, and distribute new game builds to the testing environment.
- **Localization:** AI-driven systems can assist in translating game content into multiple languages.

6. Quality Assurance:

- **Automated Testing:** AI-driven testing tools can perform regression testing, identify performance issues, and ensure consistent gameplay across different devices.
- **Playtesting AI:** AI agents can simulate player behaviors to identify potential balancing issues, exploits, or areas of the game that need improvement.

7. Marketing and Promotion:

- **Social Media Automation:** Tools can schedule and post updates on social media platforms, reaching a wider gamer audience.
- **Data Analytics:** Advanced analytics platforms can process and analyze user data, helping developers understand player behaviors, track marketing campaigns.

8. Post-Launch Support:

- **Player Support Chatbots:** AI-powered chatbots can provide immediate assistance to players, answering common questions and troubleshooting issues.
- **Live Operations:** Automation tools help schedule in-game events, updates, and content releases to keep players engaged.

9. User Feedback and Updates:

- **Sentiment Analysis:** Natural language processing algorithms can process user reviews and feedback to understand player sentiment.
- **Feature Updates:** Automated systems can facilitate the integration of new features, improvements, and content updates.

10. Monetization and Analytics:

- **In-game Purchases:** Automation can manage in-game purchases, virtual currency, and transactions.
- **Monetization Analytics:** Advanced systems can track player spending patterns and provide insights to optimize revenue strategies.

9. Security and Anti-Cheat:

- Automated Security Checks: Tools can automatically scan for vulnerabilities and potential security breaches in the game's code.
- Anti-Cheat Systems: Implement systems that detect and prevent cheating, ensuring a fair and enjoyable experience for all players.

10. Data Backup and Recovery:

- Automated Backups: Regular and automated backups of game assets, code, and databases ensure data integrity and disaster recovery.

4. How to Create Project Plan and Product Backlog for project and User Story Creation.

Step 1: Open the web browser and search for Jira Login.

Step 2: Enter your username and password, and click on the Log In button.

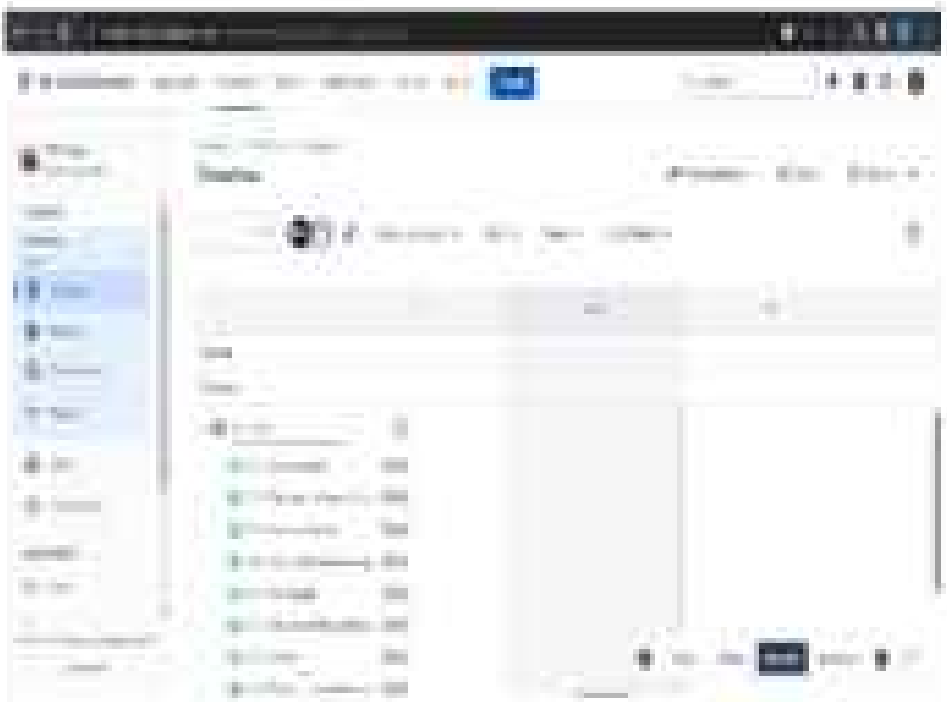
Step 3: Click on the Jira software -> projects -> create project button in the top navigation bar.

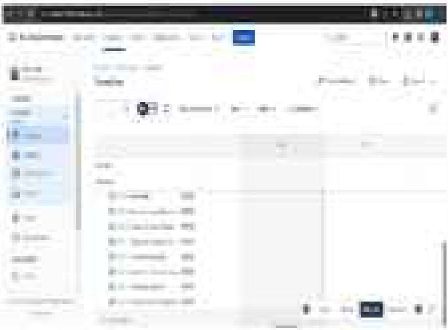
Step 4: Select Scrum, click on Use template button at the bottom.

Step 5: Give a name to your project and Give a Description if you want.

Step 6: In Those Jira creates Epic name of the project with the default user stories.

Step 7: Now go to Backlog, drag & drop the User stories and tasks and then sprint that into different Epics.





Step 8: Give the user story points to each user story and click on "Start Sprint"

Step 9: Go to Active sprint & place the User Stories accordingly in the Done State and Click on Start Sprint

Step 10: Click on the Burn Down Chart.

WEEK – 3

1. Creating Acceptance Criteria for the given User Stories by using Java Software

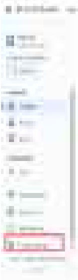
Steps For creating Acceptance Criteria

Step 1: After creating User stories with tasks in Timeline, place them in Backlog. Now go to "Issues" in the project.

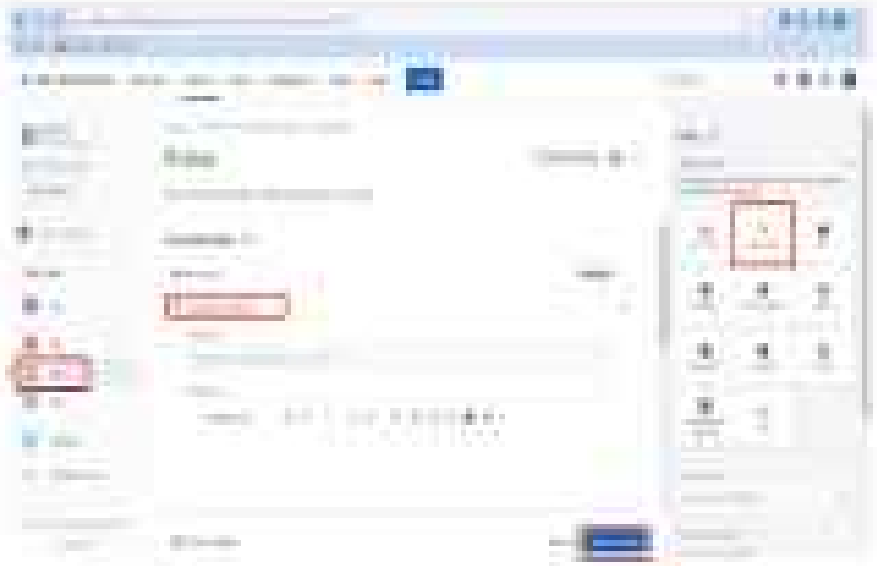
Step 2: Then select Epic and create changes as the right side of the "Create Issues" in the progress. Then create the given User stories and tasks.

Step 3: Then Go to "Issue Types" on the right side of the screen to view all the User stories and tasks of the Epic.

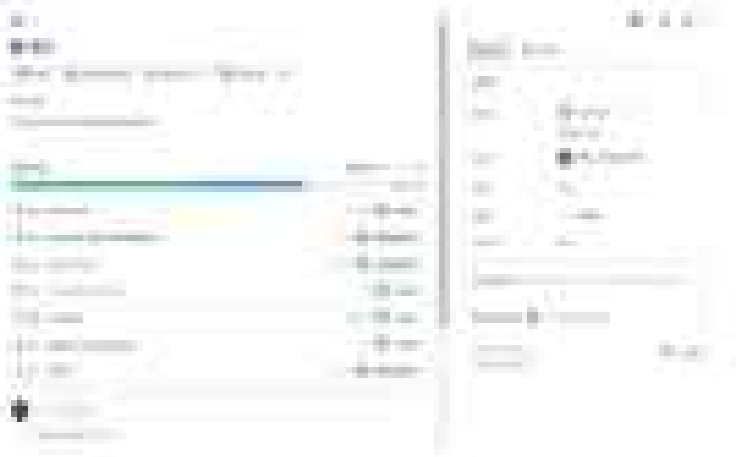
Step 6: Click "Project Settings" and then select "Issue Types"



Step 7: Now select "Story" and then select the "Description" in fields and then type "Acceptance Criteria" in the description box and then click on "Save Changes"



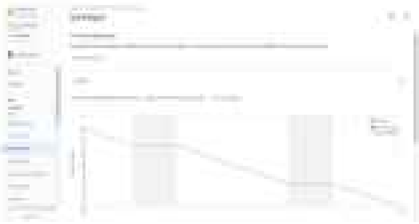
1. **Sprint Planning:** Sprint planning is an event in scrum that kicks off the sprint. The purpose of sprint planning is to define what can be delivered in the sprint and how that work will be achieved. Sprint planning is done in collaboration with the whole scrum team. Duration: In sprint, scrum encourages you to timebox sprint planning to a maximum of eight hours for a one-month sprint, with [illegible] for shorter sprints.



2. **Backlog Refinement:** [illegible] Backlog Refinement, otherwise called [illegible] Backlog Grooming, is a key process in Scrum. Backlog refinement is the process of reviewing, updating, and refining your product backlog. Backlog refinement is an important tool in your product development process because it helps your development team understand the features and functionalities that the customer wants and their requirements. Backlog refinement is an ongoing process championed by the product owner, product management, scrum masters, and representatives from the development team.

[illegible] **Burndown Chart:** The burndown chart provides an overview of how much work remains and how fast [illegible] as a project management and bug-tracking program, [illegible] communication and collaboration. Use a Burndown Chart to track the total work remaining, and to project the likelihood of achieving the sprint goal. By tracking the remaining work throughout the iteration, a team can manage its progress and respond to trends accordingly.

[illegible] **Sprint Demo:** In traditional [illegible] method, the sprint demo comes at the end of a sprint. A sprint demo is your team's chance to show off to the business stakeholders. The work that they've done is important, and you want to share it with the people who are most excited about it. The best sprint demos are celebrations. Stakeholders and developers alike are coming together to share in the work that's accomplished. They are demonstrating [illegible], new features, and [illegible] for future work.



1. **Sprint Retrospective:** The retrospective is a meeting that takes place at the end of a sprint. The length of the meeting depends on the time scale and complexity of the iteration. Retrospectives identify how to improve teamwork by reflecting on what worked, what doesn't, and why. Here team members will share all sorts of feedback whether it is Good or Bad or it can be improvement sometimes.

3. **Design Principles for UI and UX design for the created User Stories by using Wire Framing Technology like Figma.**

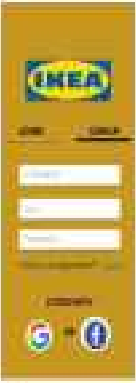
Figma: Figma is the leading collaborative design tool for building meaningful products. Seamlessly design, prototype, develop, and collect feedback in a single platform. Figma helps you gain control of complex management. You have the ability to "fork" branches of components of the design away from the main project. These forks can be edited, tested, and revised multiple times without affecting the rest of the project. When you're satisfied with the fork, you can easily merge it back into the overall design.

User Interface: The user interface (UI) is the point of human-computer interaction and communication in a device. User interface allows is the ways in which a user interacts with a piece of software, a tool, an application, or an appliance. This includes graphical elements such as menus, buttons, and other controls essential for a product's function.

User Experience: User experience (UX) design is the process design teams use to create products that provide meaningful and relevant experiences to users. UX design involves the design of the entire process of acquiring and integrating the product, including aspects of branding, design, usability and function.

1. Splash Screen

2. Login Screen



3. Homepage Screen

3. **Test case for IKEA Application**

1) Test cases for Splash screen page

Sl. No	Test Steps	Test Data	Expected Result	Actual Result	Status
1.	Create an Innovative User Interface	GUI should be presentable	Fascinating User Interface screen	Splash Screen	Pass

2) Test case for Login Screen

Sl. No.	Test Steps	Test Data	Expected Result	Actual Result	Status
1.	Providing email login option for the users	Users can login with their mobile no.	Users should be able to login with no complications	Login process should be achieved successfully	Pass

		Username or Email-Id.			
4.	Resetting forgot password option	Resetting the new password.	User should be able to reset the new password using their login details.	Resetting the new password has been achieved successfully	Pass
5.	To create a new account	To sign in into the new account using the mobile no., Email or with Google account.	To be able to create a new account and perform login process.	Cannot able to create a new account with the same mobile no.	Fail
6.	To enter into the Homepage screen	Search for interactive view of the products with its details.	Homepage should be presentable to the users	Users should be able to see the product details in a better format	Pass

3) Test cases for Homepage Screen:

S. No.	Test Steps	Test Data	Expected Results	Actual Results	Status
1.	Create a Search bar based on IOT technology	Searching of required products, materials or items	Displaying of the products based on search results	Searching of the available products in the platform	Pass
2.	To check whether all the products are displayed and arranged in a better way	Updating the advertisements displayed if any changes required	Customer should be able to see the products clearly with all the discounts offered.	Not displaying the detailed search	Fail
3.	To ensure a comparison section is used for products	Doing Comparison section based on the quality, durability and pricing of the items.	Comparison section should be displayed with various features and options.	Comparison tab should be made visible in the Homepage itself	Pass

6.	To create a user profile.	To manage and update the user's credentials.	To be made visible in the Homepage along with a small profile icon.	Displayed on the Homepage under 'My account' name.	Pass

WEEK – 3

1. **Git Client installation and setup**

Step 1: Go to web browser and Search for 'Git' and tap on the first Link.

Step 2: Now click on 'Download for Windows'

Step 3: Click on 'Click here to Download'

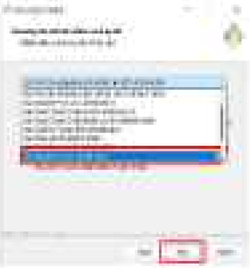
Step 4: Click on the downloaded file & click on "Next".

Step 5: Choose the path location where you want to install GIT & click on "Next".

Step 6: Enable the required checkboxes & click on "Next".

Img 7: Then click on Next

Img 8: Now select the [illegible] click on Next



Img 9: [illegible] click on

Next:

Step 10. Then click on Next:

Step 11. Click on Next:

Step 12: Click on "Next".

Step 13: Enable the 1st checkbox "Checkout Windows-style, commit Unix-style line endings" & click on "Next".

Step 14: Now enable the form option & click on "Next".

Step 15: Then enable the first option & click on "Next".

Step 16: Enable the form option & click on "Next".

Step 27: Complete the data options & click on "Next"

Step 28: Now enable both the Analytics & click on "Next"

Step 29: Then click on "Enable Sync" & click on "Next"

- Creating a repository

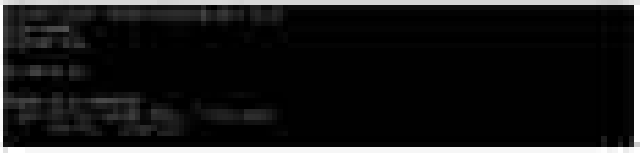
Step 1: Use the command **git init** to create a new repository

Step 2: Now enter the command **ls -al** to view all the newly added files

Step 3: Enter the command **git status** to view the status

Step 4: Enter the command **git add** to add a new or modified file

Step 5: Now enter **git status** to view the new file



Step 6: Enter the command **git commit -m "first commit"** to commit the file

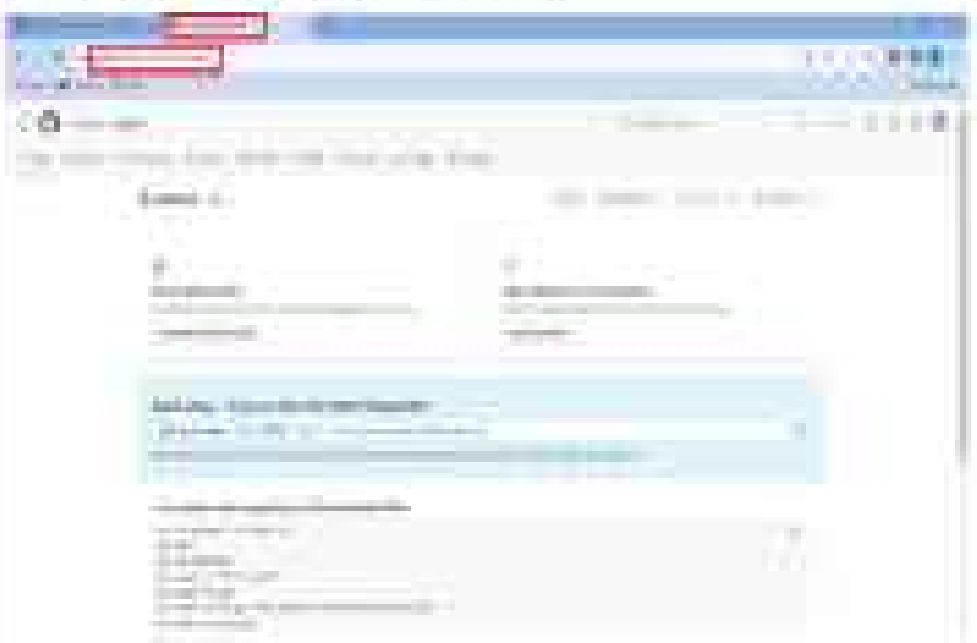
Step 7: Now enter **git config --global user.email** to add your email id and **git config**

--global user.name to add your name

Step 8: Enter **git commit -m "first commit"** to commit the file

Step 28: Then enter "git log" to view the committed files.

Step 29: Now login to the GitHub account and create a new repository named [illegible]



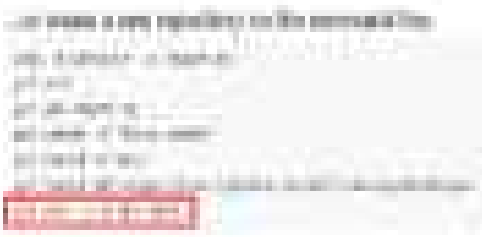
Step 34. Now copy the permalink under Link and paste it in the output

Step 35: Also copy the below command: git remote add origin https://github.com/[illegible].git & paste this link in the same notepad.

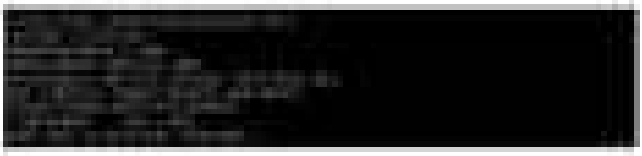
Step 36: Now copy the first link and paste it after the https://______ [illegible] the git command.

Step 37: Now paste the above created link in the Git Bash terminal & press Enter.

[illegible]



[illegible]



[illegible]

- [illegible]

[illegible]

[illegible]

[illegible]

[illegible]

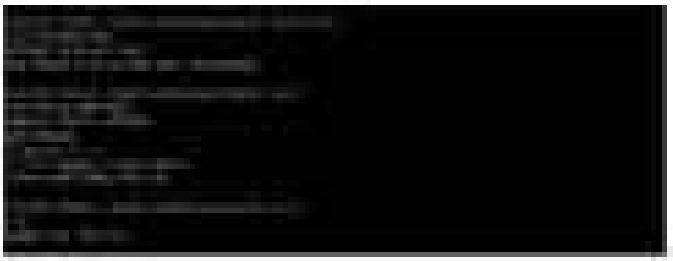
[illegible]

Step 4: Then enter `git add .` (or `git status` command)

- **Merging of branches**

Step 1: Enter `git checkout main` & type `git merge newbranch` to merge the `newbranch` with `main`

Branch. And also enter `ls` to view merged files.



- **Cloning a repository from GitHub to GitHub terminal**

Step 1: Then select any repository in your GitHub account and click on `Code`. Then copy the HTTPS

location & paste it in the GitHub terminal.

- **Working in GitHub**

GitHub: GitHub, Inc. is a platform and cloud-based service for software development and version control using Git, allowing developers to store and manage their code.

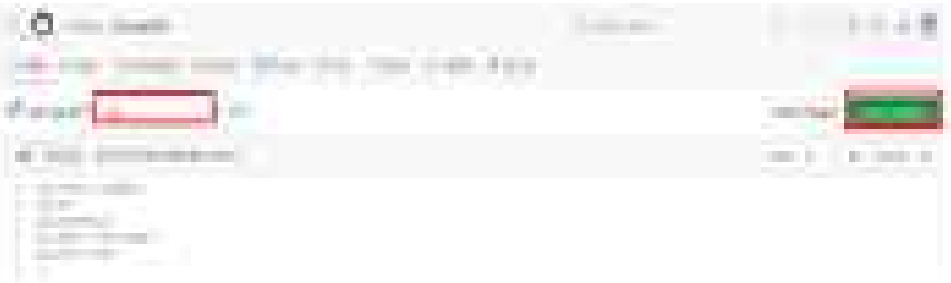
Create a Git (GitHub) account and verify/save the repository:

Step 1: Go to web browser & search for `github`. Then sign into your GitHub account. Now click on

'New' to create a new repository.

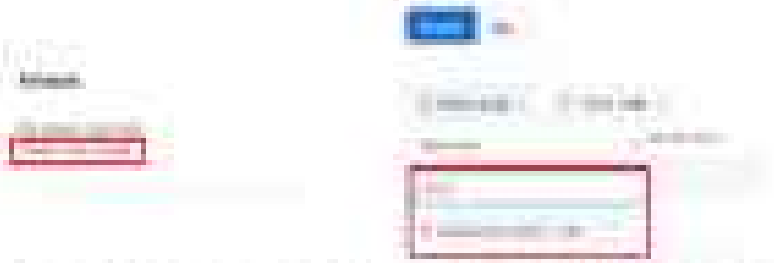
Step 2: Give a name to your repository & add description. Then add 'add a README file' and click 'Create repository'.

Step 3: Click on 'add file' and select 'Create new file' to create a new file inside the repository.



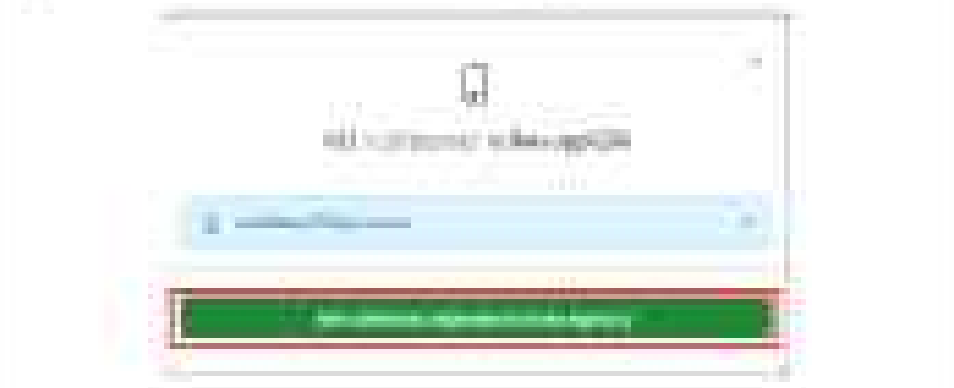
- [redacted]

[redacted]

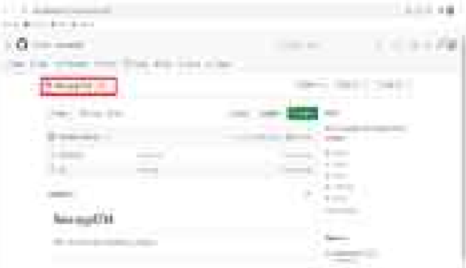


[redacted]

[redacted]



Step 23: Now the collaborator with the same project which is deployed here.



2. Create a Cloud account (AWS, GCP or any other service provider) and deploy the website.

Amazon Web Services (AWS)

Amazon Web Services, Inc. (AWS) is a subsidiary of Amazon that provides on-demand cloud computing platforms and APIs to individuals, companies, and governments, on a metered pay-as-you-go basis. These AWS [illegible] that allows a client to use basic computing

in terms of high application usage, and then scale down to reduce costs when there is less traffic.

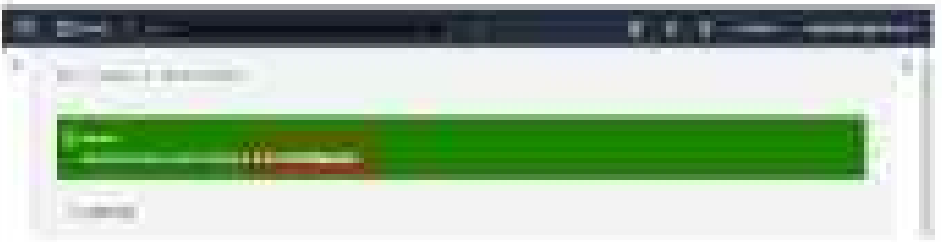
- Create and setup a virtual machine

Step 1: Go to your browser and search for AWS console and click on create a free account option to create a new AWS account by providing all the credentials required.

Step 2: After creating an account, click on "Sign in to the console" option.

Step 3: Now select the "Root User" and enter the "Email-id" & "password" which is registered with. Now you are ready and have access to "Aws".

Step 4: In the AWS home tab, scroll down and click on "Launch a virtual machine".



Step 11 Select the text you want to translate...

Step 12 Click on the translate button

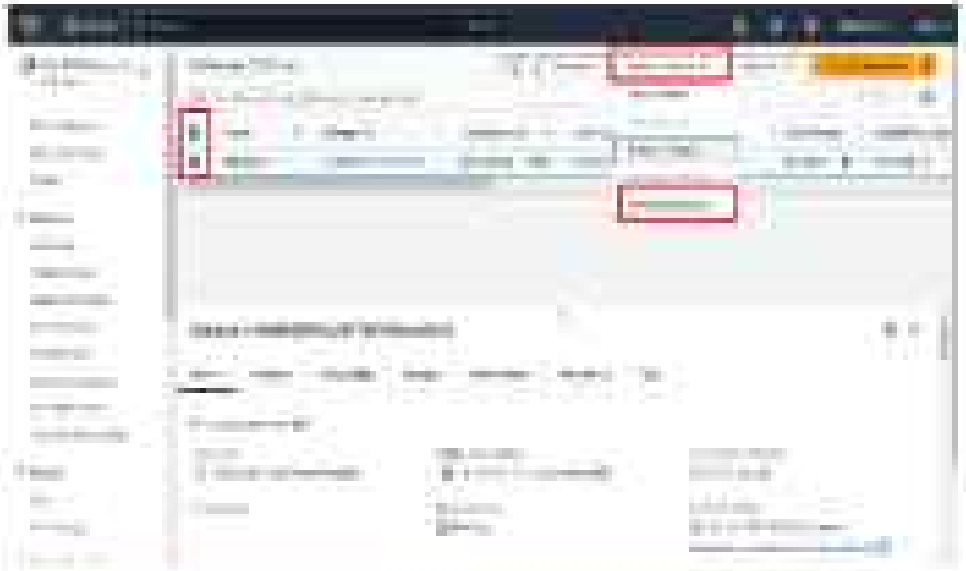
Step 13 Then click on the desired target language and wait for the translation...

Step 17: Click on Connect.

Step 18: Now open the password which you copied in step 5 and click on ok.

Step 8: Click on Yes

Step 9: Now the Firewall configuration will be completed successfully



8. Create a simple Web application using Cloud Services.

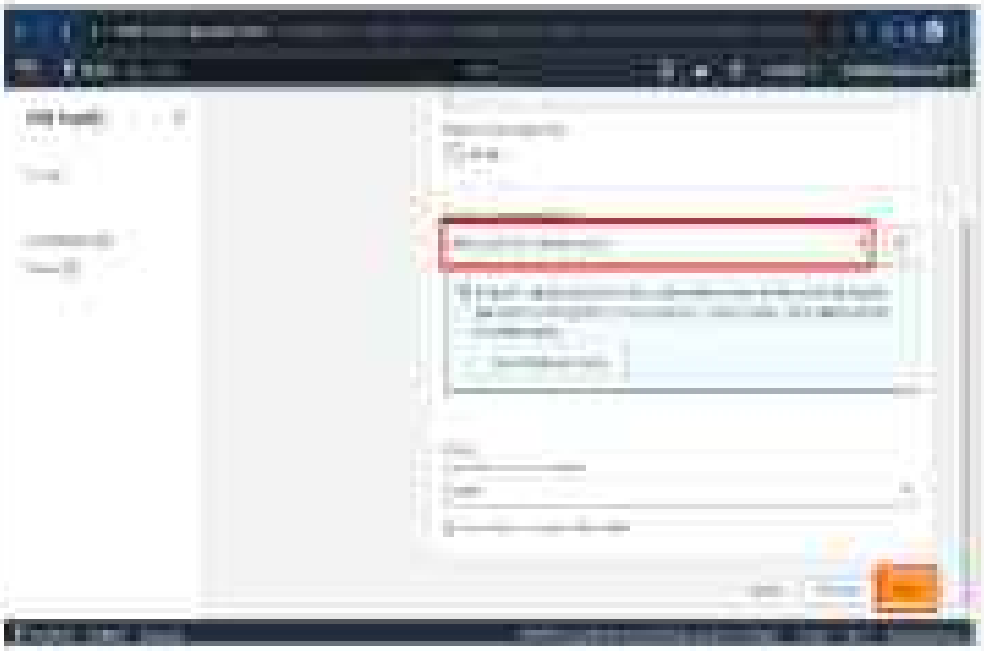
Step 1: Sign up to the GitHub account which was registered to AWS account. Then click on the link https://github.com/login/oauth/authorize?client_id=... which is used to create a new Task File.

Step 2: Click on 'Create Task'

Step 3: Now go back to the AWS Console and click on the refresh button.

Step 26: Then click on "Install & Authorize".

Step 27: Select the repository and click on "Save".



Step 06: Navigate to Custom checkbox and click on 'Next'

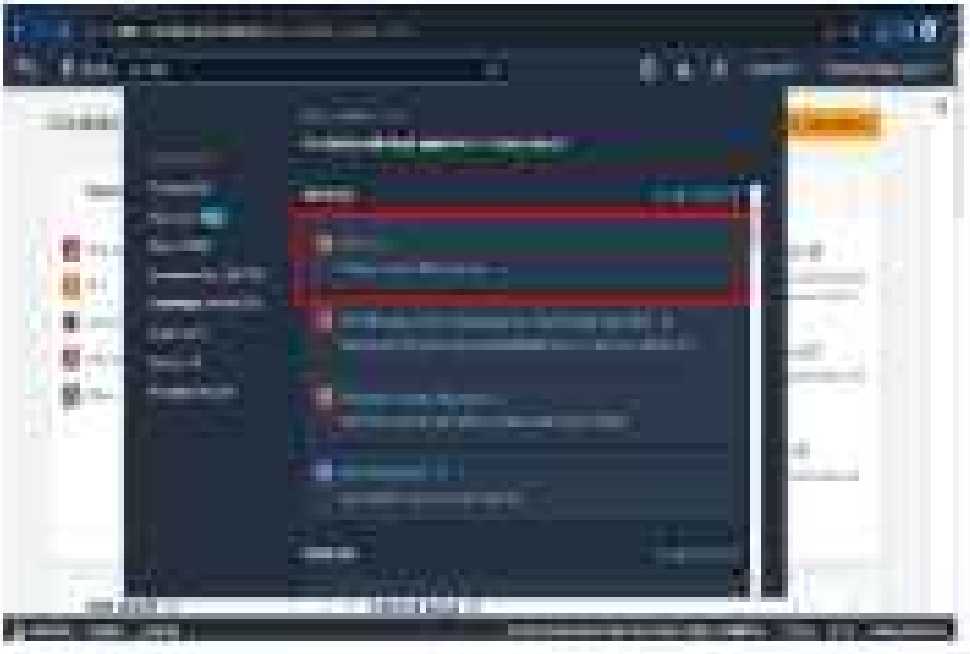
Step 07: Click on 'Submit'

Step 30: Wait until the Provision, Build and Deploy process completed. And to view the web app, click on the below generated link.

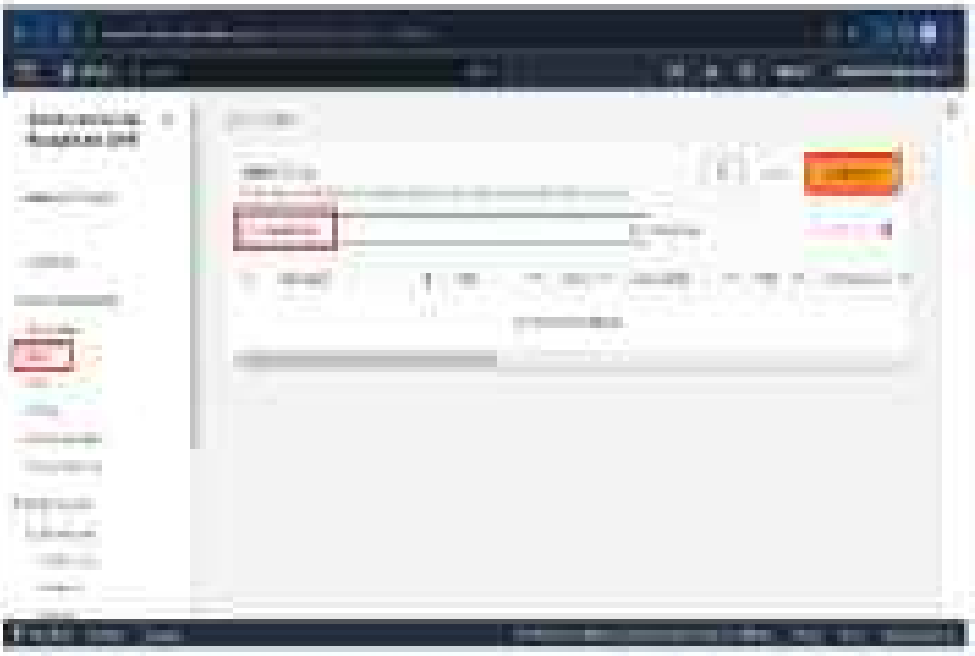
Step 31: Here we can see the web app which we have developed.

C. [illegible]

Step 2: [illegible]



Step 3: [illegible]



Step 6: You can also change your old password and set a new password and then click on **Confirm password change.**

3. <u>Create a build pipeline for simple web application such as To-do app, BMI calculator, Number converter, Word Count etc.</u>

Step 1: Go to the Web browser and open the localhost:8080 (Jenkins port number) and give entry.

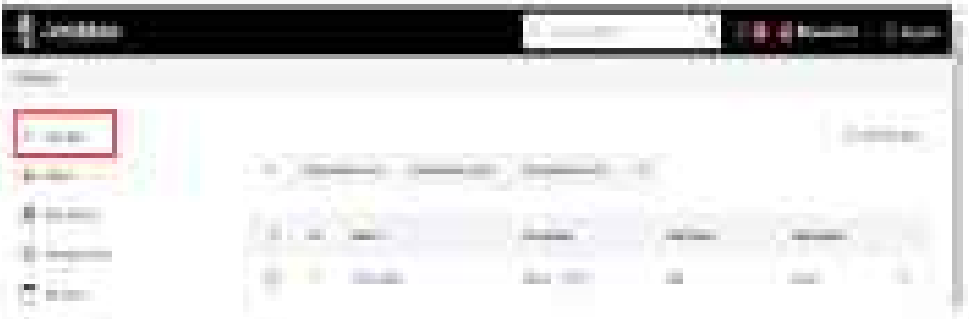
Step 2: In the Jenkins Dashboard → Click on new item and create a free style project named "Wheat"

Step 3: On the next page, click the option and scroll down to the **build steps**

Step 4: Select "Execute **windows batch** command"

Step 5: Add static command and click on Save

Step 6: Now go back to dashboard and create another simple project named "Sample Project".



Step 7: Give any description in order to Build Trigger tab. Select Build after other projects are built

and select previous job created i.e. Build1.

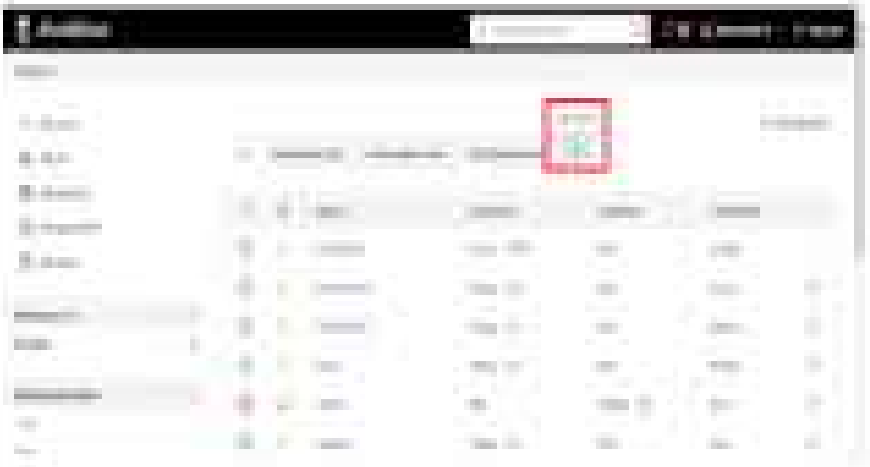
Step 8: Create three more Jobs creating another three Freestyle projects.

Step 9: After the completion of all four projects, Go back to dashboard and select Manage Jenkins.

Step 20: Go to **Plugins**.

Step 21: Click on **Available Plugins** and search for and install the **polylang** plugin.

Step 22: Now Go back to the **Dashboard**. Click on **Icons**.

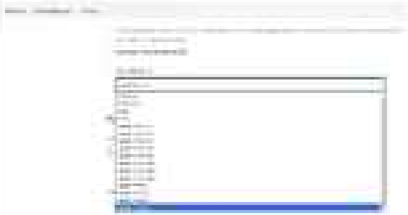


Step-13: On the next page, give a name to the pipeline and select a **Build Pipeline** View and Click on **OK**.

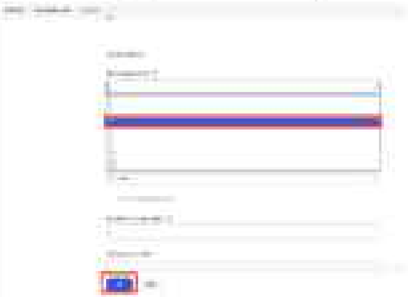
Example

Step-14: Now add the description and scroll down to select **Initial job** section.

Step 25: Select the Initial Job as [illegible].



Step 26: Scroll down to **Display Section Tab** and Select the **no of Displayed builds to be 5**, click on **ok**.



Step 27: Click on **Trigger a Pipeline** to View the Pipeline.

Step 28: Here you can see the pipeline BUILD View of all the jobs.

Step #1: All the successful jobs are Green in the graph, pending in blue and failed in red. Now click on

Console Output to view the build results.

WEEK – 4

1. HTML, CSS and Java Script Fundamentals (Code structure – statements, comments, variables, constants, data types, interaction, operators, comparisons, control flow, functions)

A) Using HTML with Javascript

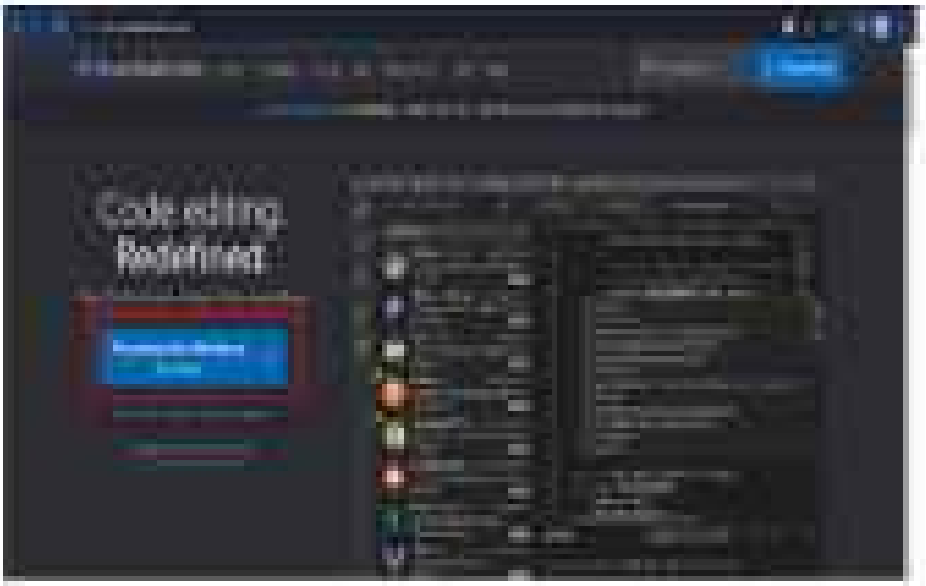
B) Using CSS file

3. Setting up the Environment and tools for Front End Development

- Installation of VS Code

Step 1: Go to Web Browser and search for VS Code. Then click on the first link.

Step 2: Click on "Download for Windows".



Step 3: Double click on the downloaded "VSCodeUserSetup-x64-1.63.2.exe" file.

Step [illegible]: Choose the location to save the file & click on 'Next'.

Step [illegible]: Select a folder name & click on 'Next'.

Step [illegible]: Enable all the checkboxes and click on 'Next'.

Step 7: [illegible]

Step 8: [illegible]

Step 8: Go to Desktop > Right click > New > Folder > Folder name.

Step 9: Launch the VS Code > click on File > Open Folder > Select the newly created folder.

- VS Code extension:

Step 10: Click on Extensions logo and type for Live Server > Then click on Install to install the server.

- Create a single form like Registration form, feedback form, after submit hide create form and enable the display section.

Registration Form

Identity Form:

Final Output:

5. Create and run simple program in Type Script.

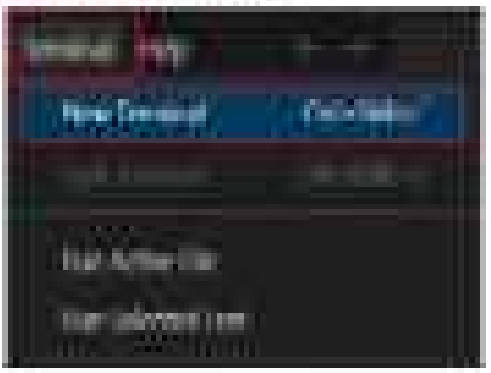
- **Installation of Typescript using Node.js Package Manager (npm).**

Step 1: Go to web browser & search for Node.js. Then click on 'Download' and download the latest version.

Step 2: Now enter the below commands in the Command Prompt to check the current installed version of Typescript.

- Installation of typescript inside the IIS node.js server:

Step 1: Launch your VS code -> Terminal -> New Terminal.



Step 2: Select the Terminal as 'Command Prompt'

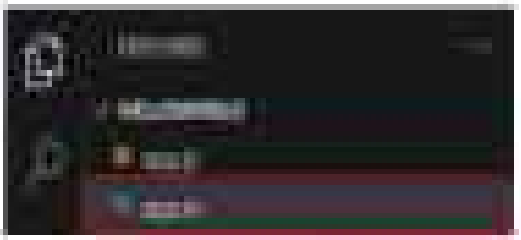
Step 5: Enter `npm install typescript --save-dev` to install the typescript dependency.

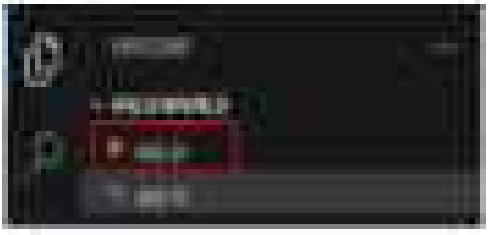
Step 6: Enter `npm install typescript -g` & `npm install typescript@next -g`.

Step 7: Enter `npm install -g live-server` to install the Live Server.

Step 8: Enter `live-server` to run the Live Server.

Step 9: Create a new empty Folder in the Desktop & open that folder in the VS Code. Then create a new file named `app.ts` & write the below code.





Step 19: Then make changes to the app.ts code as shown below

Step 20: Enter tsc app.ts to run the compiler

Step 21: Now go to the index.html file > Right-click on it > Select **Open with Live Server**

Output:

1. [illegible]

A. Login.html

B. Logika

C. Laporan

Output:

Please leave some space here while writing in record.

And continue from here

4. Creating a repository in Github and cloning the repository in Vs code.

Step 1: Create an empty folder in the Desktop and open that folder in VS Code. Then create a new file named 'app.java' & add some code to it.

Step 2: Go to Extensions -> Install Git Repository

Step 3: Now click on ▾ next to

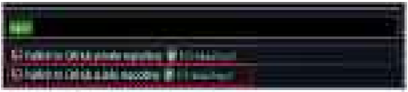
Step 4: Enter a commit message and click on

Step 6: Then click on "Publish Branch".

Step 8: Now again login your GitHub account and click on "Sign in".

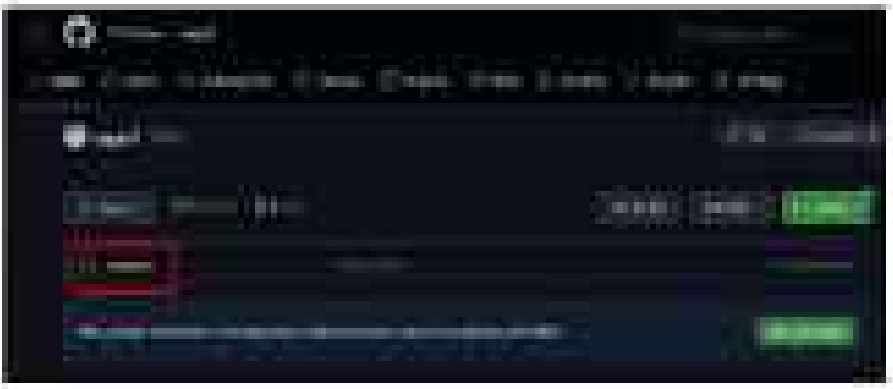
Step 9: Now click on "Authorize Visual Studio Code".

Step 8: Select "Publish to GitHub public repository".



Step 9: Click on "Open on GitHub".

Step 10: Now you can see the newly file [illegible]



Step 21. Then click to create a pull request.

- Synchronizing Changes to the code

Step 22. To sync changes to your code, click on ‘Commit’.

Step 23. Type in a commit message and click on it.

Step 24. Click on ‘Sync Changes’.

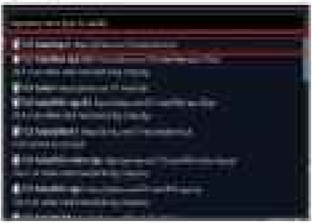
Step 05: The changes make are being displayed here.

- Cloning an empty repository from VS Code to GitHub

Step 01: Click on 'Clone Git Repository'.

Step 02: Now click on 'Clone from GitHub'.

Step 18: Select a repository you want to Clone.



Step 19: Now open the selected repository in your GitHub Repository.

Step 20: Here, I've selected the already created repository for displaying the sample Hello World java program as shown below.

WEEK – 3

1. Creating and running a React.js app using VS Code

Step 1: Create a new folder in the Desktop & open the folder in Visual Studio Code.

Step 2: Navigate to Terminal → New Terminal → Select Command Prompt.

Step 3: Enter the command npm init to install a new package.

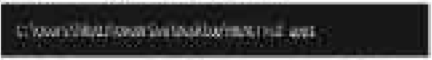
Step 4: Enter npm -v to check the installation of Node Package Manager and the version on your Windows cmd.

Step 5: For installing, make sure to check the installation of Node.js by entering the command "node -v" to check the current version installed.

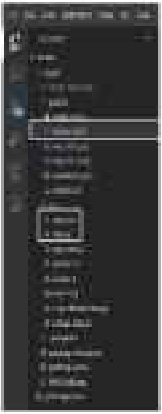
- **Folder structure for React App**

Step6: Enter the command 'npx create-react-app app1' to create a React Project & give the folder name as **app1**.

Step7: Enter the command 'cd app1' to change the working directory.



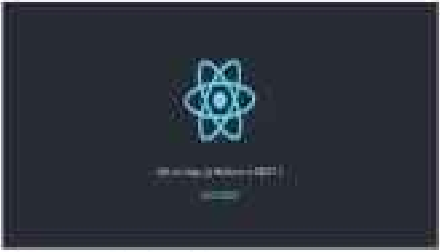
Step8: Here under the 'node_modules' then public & Select the 'app.js' and 'app.css' file from the src folder on the right side of your VS Code Window.



Launching of React.js app

Step9: Enter the command 'npm start app1' to run the react app.

Output:



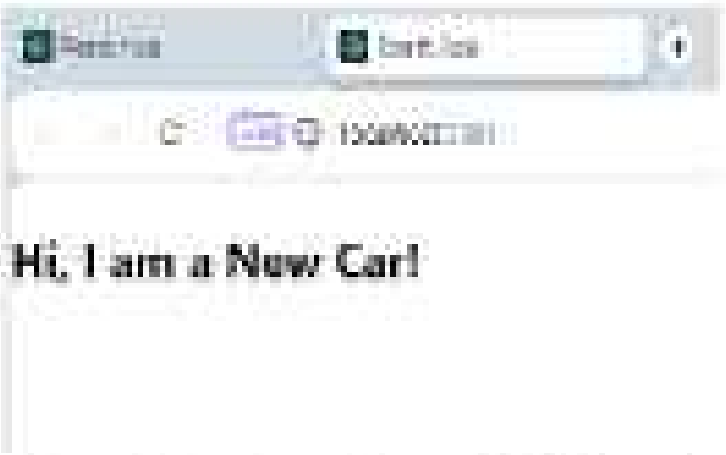
Creating your own changes to the code:

1. Root Component

   Functional Components: Functional components are simply javascript functions. We can create a functional component in React by writing a javascript function. These functions may or may not receive data as parameters. The functional component is also known as a stateless component because they do not hold or manage state.

Hi, I am a Car!

Class Component: A class component can hold more complex functionality than functional components. A class component must include the extends React.Component statement. The class component is also known as a stateful component because they can hold or manage local state.



React Props: Props is short for properties and they are used to pass data between React components.

React State. React has another special built-in object called state, which allows a component to create and manage its own data. Unlike props, a component cannot pass data via state, but they can create and manage it internally.

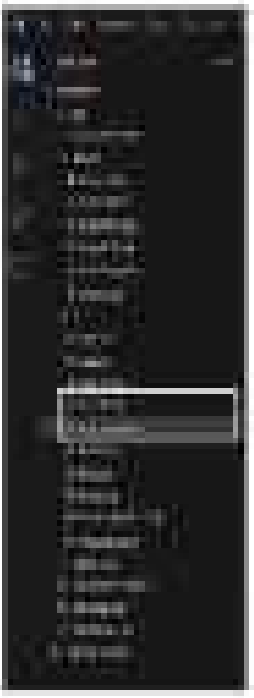
I'm from Karnataka State.

Render a List in React with Keys. In the following React List example, we render a list of items. And, we include a unique identifier assigned to `id`. We are using the `map()` method to iterate through the `numbers` array and return a unique key property. Keys are used in React to update a list, such as adding, removing, or moving an item in a list.

1. Testing React App using Jest

Step1: Create a new react app using create-react-app react-testing .

Step2: Create a component called <Counter /> in the



Step1: Now open the main.py file in the src folder and do the below changes as shown below.

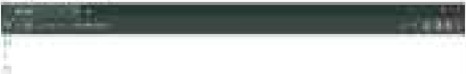
Step2: Then go to Terminal _ New Terminal _ Select Command Prompt & give the command "npm run android" to run the android file.

Output:

Step: Run to your simple flask application

Increment Value:

Decrement Value:



4. **Testing single page application (Registration form) using React:**

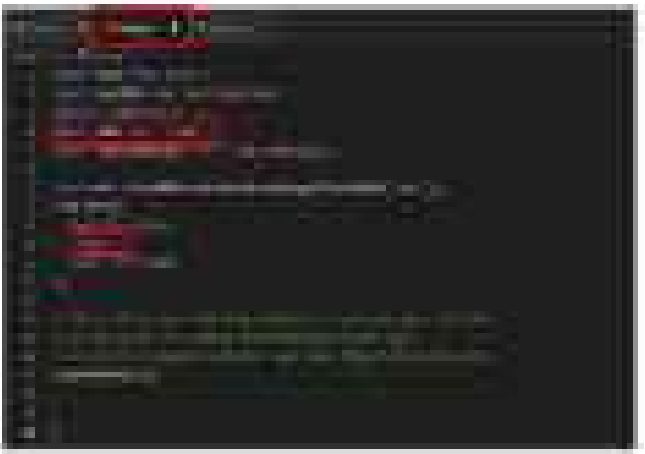
Note: [illegible]

Step1: Create a new react app 'npx **create-react-app** app1'

Step2: Now create a new file named 'Reg.js' in the 'src/' folder and write the below code

Step 2: Now create a new file named "app.py" in the root folder of your application and add the following code.

Step 3: Now create a new file named "index.js" in the root folder of your application and add the following code.



Output:


User Registration


User Registration
User Details successfully registered!

WEEK - 6

1. Implement navigation using react router.

Components in React Router:

There are two types of router components:

- **<BrowserRouter>**: It is used for handling the dynamic URL.
- **<HashRouter>**: It is used for handling the static request.

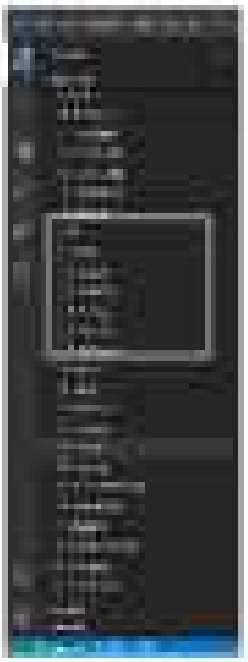
Step 1: Open the VS Code and launch the React or the server application.

- **Add React Router:**

To add React Router in your application, run this command in the terminal. Make sure this is inside your React application folder.

Step 2: Open the folder in VS Code and in the 'src' folder, create a new folder named 'Pages'.

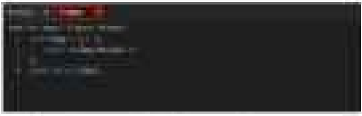
Step 3: Now, within the 'Pages' folder, create the new files named 'Blogs.js', 'contact.js', 'home.js', 'Layout.js' & 'NoPage.js' as shown below:



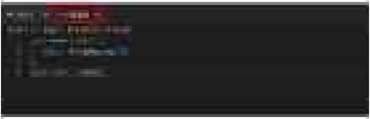
Step 4: Then add some code inside the 'index.js' file as shown below:

Step 5: Now add code to "[illegible]" file to style your application.

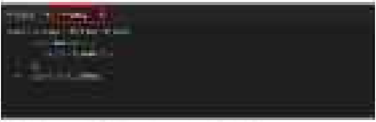
Step 6: Write the below code to "[illegible]" file inside the "Pages" folder.



Step 7: Write the below code to "[illegible]" file inside the "Pages" folder.

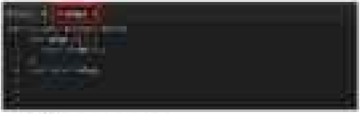


Step 8: Write the below code in 'Home.js' file inside the 'Pages' folder



Step 9: Write the below code in 'Layout.js' file inside the 'Pages' folder

Step 10: Write the below code in 'NoPage.js' file inside the 'Pages' folder



Now run the 'index.js' file to view the final output.

Output:

2. React Hooks: Introduction, Rules of Hooks, useState, useEffect, useContext, useReducer

React Hooks: React Hooks are a new feature introduced in the React 16.8 version. It allows you to use state and other React features without writing a class.

- Types of Hooks in React

1. UseState - The useState Hook allows you to create and manage state in a functional component. State represents values that can change over time, such as user input or data fetched from an API. The useState Hook returns an array with two elements: the current state value and a function that can be used to update it. This function is typically named setSomething, where Something is the name of the state value.

   Syntax: const [value, setValue] = useState('initial');

2. UseEffect - The useEffect Hook allows you to perform side effects in your components. Some examples of side effects are: fetching data, directly updating the DOM, and timers. useEffect accepts two arguments. The second argument is optional.

   Syntax: useEffect(<function> , <dependency>)

3. UseContext - React Context is a way to manage state globally. It can be used together with the useState Hook to share state between deeply nested components more easily than with useState alone. To use useContext, you must first import createContext and useContext in

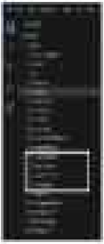
   Syntax: const myContext = useContext(MyContext);

4. UseReducer - The useReducer Hook is similar to the useState Hook. It allows for custom state logic. The useReducer Hook accepts two arguments.

   Syntax: useReducer(<reducer>, <initialState>)

Step 1: Open the VS Code and launch the React.js application.

Step 2: To run the React Hooks, create the files [illegible] in the 'src' folder.

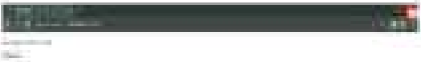


- **Example of UseState:**

Step 3: Write the below code in the app.js file inside the 'src' folder.

Step 4: [illegible] file within the 'src' folder.

Output:



- **Example of UseEffect**

Step 1: Write the below code in the App.js file inside the src/ folder.

Step 2: Then add code to [illegible].js file within the src/ folder.

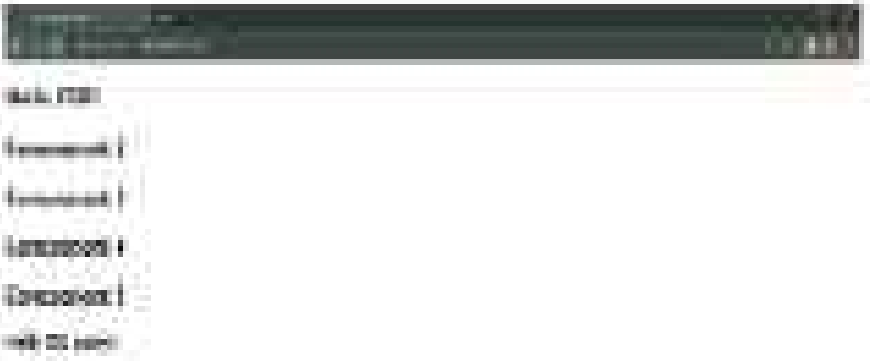
Output:

- Example of UseCounter:

Step 7: Write the following code in the file src/App.js (Replace the entire file with this):

Step 8: Then add the code in the file src/App.css (Replace the entire file with this):

Output:

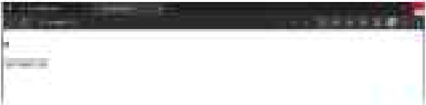


- Example of Unification:

Step 1: Write the below code in the app.py file inside the src folder.

Step 2: Execute code in the terminal as python app.py in the src folder.

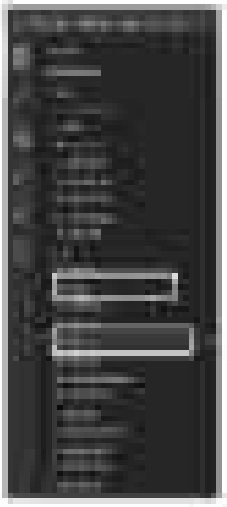
Output:



3. **Build single page application - Shopping cart**

Step 1: Open the VS Code and launch the React in the local server.

- **React folder structure:**



Step 2: Add routes in `routes.js` file in the `server` folder.

Step 3: Take routes in the `index.route.js` file in the `routes` folder in the `server` folder.

Output:

4. Setting up the environment and tools to install Java Runtime stable version and add environment variable to it.

Java: Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let programmers write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need to recompile.

Installation of Java on Windows:

Step1: Go to Web browser → Search for Java → Click on the first link shown

Step2: Now click on "Download Java".

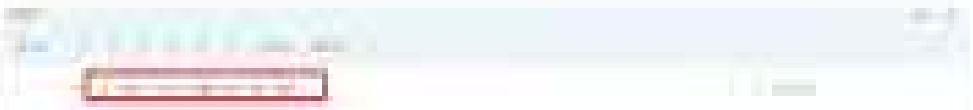
Step 5: Wait until the installation is completed.

Step 6: Click on "close".

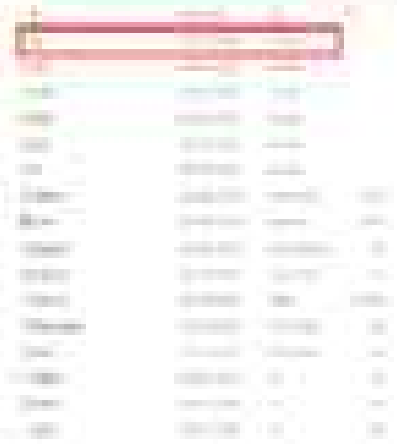
- **Setting up of Environment Variables for Java.**

Note: To run Java programs, make sure to setup the Environment variables for Java on your windows system.

Step 1: After the installation of Java, go to This PC → C drive → Program Files → Java → [illegible] as shown below.



Step 2: Now click on Your Slide:



Step 3: Then copy the copy the location here:

Step 4: Go to database and paste here :- Control Panel\System and Security\System\Advanced system settings

Step 5: Now click on **Environment Variables** button.

Step 6: Under **System Variables**, select **Path** location & click on **Edit**.

Step 7: Click on **New**

Step 8: Now paste the bin location which was copied in step 3.

Step 9: Use Command Prompt to check whether the Java Program is working or not.

Give the command as 'java -version' to check the Version and other details as shown below.

Step 10: Now again enter the command as 'javac' to check the working of Java compiler on your Machine.

3. Install DBMS (MySQL, PostgreSQL, or any other)

MySQL: MySQL is an open-source relational database management system (RDBMS). SQL is a language that programmers use to create, modify, and extract data from the relational database, as well as control user access to the database. It manages the store, allows for retrievals access, and facilitates storing database integrity and crash recovery functions.

- **Installation procedures for MySQL:**

Step 1: Go to Web browser → Search for MySQL → Click on **Downloads**.

Step 2: Click on **MySQL Community (GPL) Downloads**

Step 3: Now click on the MySQL Installer MSI Windows.

Step 4: Then select the second option to click on Download.

Step 5: Now you can Login or Sign up with your Oracle account or simply click on No thanks, just start my download.

Step 6: Validating the license agreement

Installing the MySQL workbench and server administrators

Step 7: Select "Custom" & click on "Next".

Step 8. Select the SQL/SSIS products from **Available products** to **Products to be installed** and then click on "Next".

Step 9. Click on "Execute".

Step 10. Enable the option below & click on "**Install**".

Step 11. After the installation of the products are complete, click on "Next".

Step 12: Click on Next.

Step 13: After the completion, click on Next.

Step 12: Click on 'Next'.

Step 13: Once changes are saved successfully, click on 'Next'.

Step 16: Click on 'Next'.

Step 17: Create a strong password for WSE network data access user.

Step 5B: Click on the 'Next' button.

Step 5C: Click on the 'Next' button.

Step 20: Click on 'Courses'.

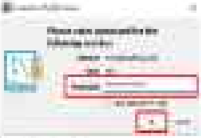
Step 21: Scroll down to 'Next'.

Step 22: Click on 'Next'.

Step 23. Click on **Finish**.

Step 24. Now click on **Local instance MySQL80** icon.

Step 25: Now enter your MySQL Server password here & click on "OK".

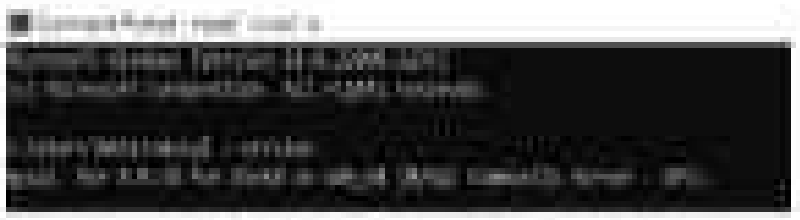


Step 26: Here is your MySQL Workbench is ready to use.

To run MySQL server:

Going the Environment Variables for MySQL server & enter the command as "mysql -u root -p". Now enter your password to run the MySQL Monitor.

Step 1: Go to command Prompt → give the command as "mysql --version" → to check the current version of MySQL.



Step 2: Now give the command as 'mysql -u root -p' & enter the password to execute the MySQL server.

Step 3: Type the command 'show databases;' to view the existing databases.

Step 4: Enter the command 'create database employee;' to create a new database.

Step 5: Again enter the command 'show databases;' to view the created database as shown as below.

WEEK – 1

1. Installation of Eclipse IDE for Enterprise Java and Web Developers latest version on Windows.

Step 1: Go to Web Browser → Search for Eclipse download → Click on the first link as shown below

Step 2: Now click on 'Download x86_64' option

Step 3: Then click on 'Download'


eclipseinstaller

Step 6: Select click on Next.

Step 7: Click and the installation is complete.

Step 6: Now click on **Commit**.

Step 7: Now the Eclipse IDE is ready to use.

1. Create Spring application with Spring Initializr using Annotation like Spring Web, Spring Data JPA.

Step 1: Go to web browser → search for "Spring Initializr" & click on the first link

Step 2: Then select the below options.

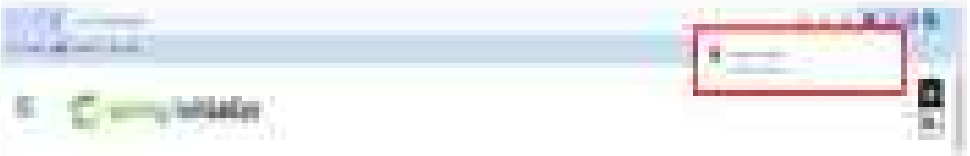
Step 3: Unzip the project file & open the project in "eclipse/sts".

Step 6: Now click on 'ADD SHIPPING ADDRESS'.

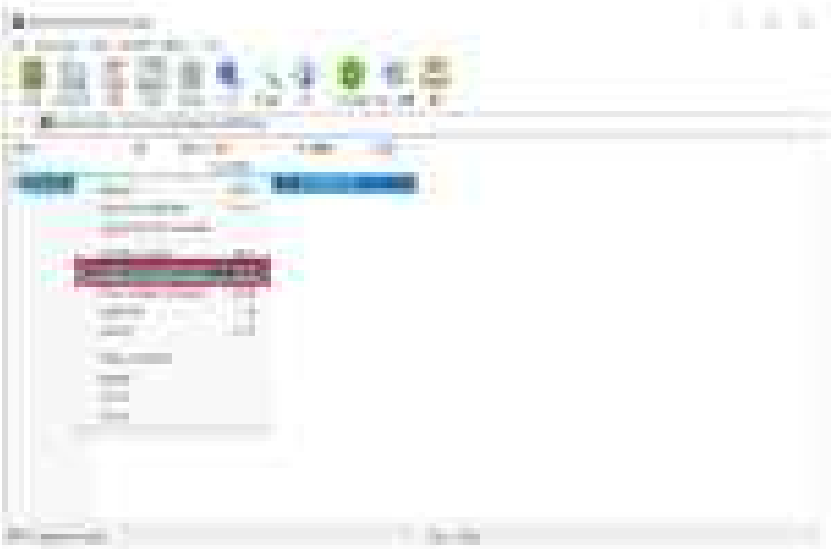
Step 7: Under the 'Mode' section, select the option 'Pickup'.

Step 8: Now click on 'GENERATE' option.

Step 9: Click on the zip file downloaded.



Step 5: Now remove the zip file which was downloaded.



Step 6: Now Right Click on the Recycle Bin & click on Yes.

Step 11: Now launch the Eclipse IDE, go to **File → (Scroll to Import)** option...

Step 12: Then select the **Existing Maven Projects** option from the **Maven** folder & click on **Next**.

Step 13: Click on **Browse** to specify the Root Directory.

Step 12 Now select the Location of the file for where the .zip file was extracted.

Step 13 Now click on **Finish**.

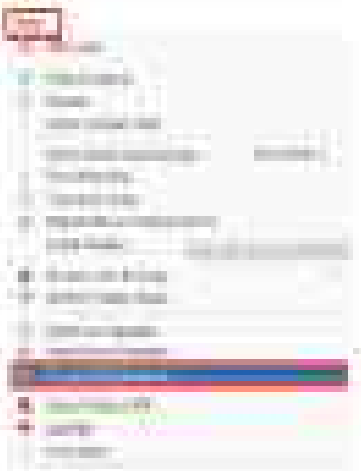
Step 14: Now go to → [illegible] → src/main/java → [illegible] in the Package Explorer window.

Step 15: Then add the below code in that file & Save it.

Step 13: Now right-click on the `springboot-application.java` file → **Run As** → **Java Application**. If the option is not visible, click on **"Other..."** and select **Java Application**.

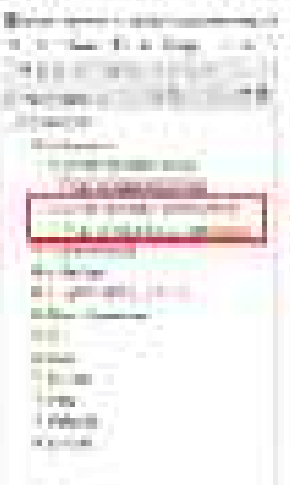
Output:

Step 14: Go to **Help** → **Eclipse Marketplace** and search for and install the required plugin tools.



Step 20. Now search for 'spring tool suite'. It should be 'Spring Tool Suite (spring tool suite) 4 (3.9.0 RELEASE)' version.

Step 21. Now create a new package named 'com.springboot.springboot.controller' & also create 'springbootApplicationController.java' file in the 'Project Explorer' section.



Step 22. Then write the below code in the 'springbootController.java' file & then save it. Now run the 'springbootApplication.java' file as JAVA APPLICATION.

Step 22: Then go to the Web browser, type localhost:8080/hello/ & press Enter.

Output:

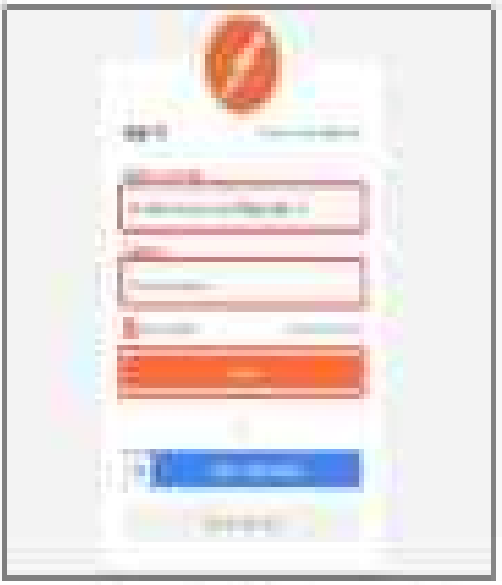
WEEK – 8

1. **How to create POSTMAN version & Install Postman**

Step 1: Go to Web Browser _ search for "Postman" _ select the second option there

Step 2: Then click on "Windows 64-bit" option to download for windows OS.

Step 3: You can sign up / sign in your account by giving your Email-id and Password here.



- Test created API with the help of Postman

HTTP Methods

- **HTTP GET**: Use GET requests to retrieve resource representation/information only – and not to modify it in any way. As GET requests do not change the state of the resource, these are said to be safe methods.
- **HTTP POST**: Use POST APIs to create new subordinate resources, e.g. a file is subordinate to a directory containing it or a row is subordinate to a database table.
- **HTTP PUT**: Use PUT APIs primarily to update existing resource. If the resource does not exist, then API may decide to create a new resource or not.
- **HTTP DELETE**: As the name applies, DELETE APIs delete the resource (identified by the request URI). A successful response of DELETE request should be HTTP status 200 (OK) along with a response body, perhaps representing a deletion confirmation message (using suitable representation), or using the HTTP status code 204 (NO CONTENT) with no response body.

Step 1: Now click on 'Create Collection' button

Step 2: Log in to Cisco's Official Site.

Step 3: Give a name to your calendar & click on 'Create' option from the bottom.

Step 6: Scroll and click on "Save"

Step 7: Click on "Update"

Step 8: Now select "GET" method & click on the HTTP Link & then click on Save

Step 7: Now go to → Tests → Enter the below code → Save & Send

Step 8: Then check the response status here. If the status is 200 OK, means the response has been accepted by the POSTMAN.

Enter the below code → Save it & Send.

Now perform the same procedure for other two HTTP methods as shown below.

Step 14: Now select "PUT" method & click on the HTTP link as shown below & then go to → Tests → Enter the below code → Save it & Send.

Step 15: Now select "DELETE" method & click on the HTTP link as shown below & then go to → Tests → Enter the below code → Save it & Send.

Step 15: Save the TEST - Basic results report as shown below.

3. Create REST controller for CRUD operations Employee Service REST API

Step 1: Open Eclipse IDE. In Enterprise edition, click on File menu → New → Project... option as shown below.

Step 2: [illegible]

Step 5: Give the Name as "User" → select "Maven" → Description as **User Registration Spring Boot Application** → Package as com.example.user & then click on "Next"

Step 6: Now add all the required dependencies as shown below & click on **Finish**.

Spring REST – creating Spring REST controller

- Controller Layer (Handling request and response)

- Service Layer (Application business logic)

- Repository Layer (Communicating with DB)

Step 7: Then select file new folder ... New ... Package

Step 6: Create a new package named `com.example.app.controller` in `src/main/java`.

Step 7: Create a new package named `com.example.app.repository` in `src/main/java`.

Step 8: Create a new package named `com.example.app.service` in `src/main/java`.

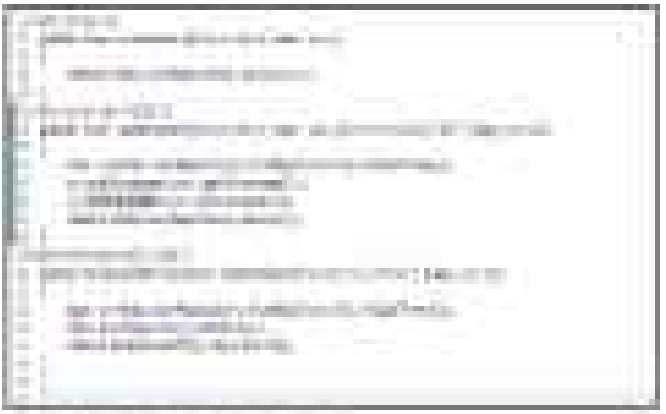
Step 11: Now right click on the "com.example.demo.controller" package -> New -> Class & create a new class named "UserController" & then click on Finish.

Step 12: Now right click on the "com.example.demo.entity" package -> New -> Class & create a new class named "User" & then click on Finish.

Step 13: Now right click on the … package > 'New' > Interface to create a new class named '…Repository' & then click on 'Finish'.

Step 14: All the created packages and classes are displayed here.

Step 5: Now add code in UserController.java file as shown below



Step 6: Now add code in User.java file as shown below

Step 17: Now add code to **UserRepository.java** file as shown below:

Step 18: Now go to the 'src/main/resources/' folder -> select 'application.properties' file here.

Step 19: Now add code to **application.properties** file as shown below:

Step [illegible]: [illegible] **DemoApplication.java** [illegible]

Step [illegible]: [illegible] **DemoApplication.java** [illegible] **Run as** [illegible] **Spring application**

Step [illegible]: [illegible]

- Creation of database using MySQL Command Line Client

Step 23: Open the command prompt ↵ give the command as **show databases;** to view all the exist databases ↵ now to create a new database give the command as **create database lakshmi;** to create a new database with as name ↵ again enter **show databases;** to view the newly created database as shown below.

- Connection of newly created database to the MySQL Workbench

Step 24: Go to ↵ option to add new connection.


Welcome to MySQL Workbench

Step 27: Now click on "OK" option as shown below.

- Creation of simple table

Step 28: Now write the below code to create a simple table as shown below.

Note: To run the code, select each code line separately & press **ctrl+shift+enter** key to view the output.

- Testing using CRUD operations using Postman RESTful web services

Step 28: Login to the Postman & click on 'Create collection' button

Step 29: Again click on 'Create collection' button

Step 30: Give a name to your collection & click on 'Create' option button

Step 36: Navigate to MySQL Workbench & enter the below command as shown below. To run the code select and click on [illegible] & press **ctrl+shift+enter** keys to [illegible] the output.

Note: Here the POST method is executed successfully.

Step 37: Select POST method & then enter the [illegible] localhost:8080/users/ [illegible] → select **Body** → Text type → JSON → enter the below code as shown below → [illegible]

Step 38: Now go to MySQL Workbench & copy the below command as shown below. To see the table [illegible] separately [illegible], press **ctrl+shift+enter** key to view the output.

Note: Now the PUT method is updated successfully.

Step 39: Select 'DELETE' method & then enter the command with the id to operate [illegible] localhost:8080/[illegible] → select **Body** → Text type: JSON → enter the below code as shown below → [illegible] Send.

Step 10: Now go to MySQL Workbench & enter the below command in a new tab/file. To run the code select each code line separately & press **ctrl+shift+enter** key to view the output.

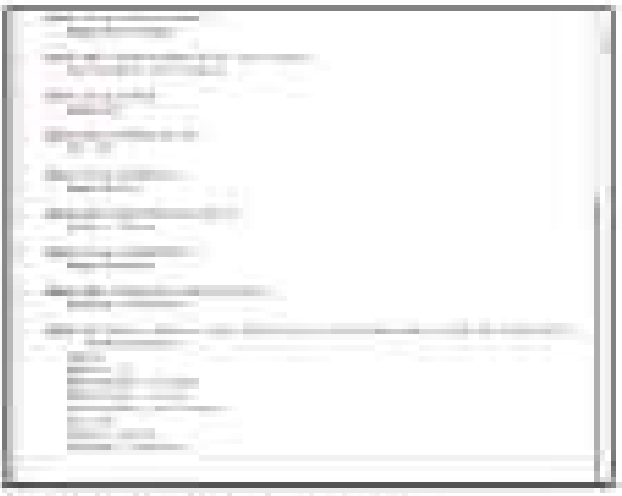
Note: Here the SELECT method is optional to execute.

3. <u>Create user registration form.</u>

Step 1: Now go to Eclipse IDE for enterprise → File → New → Spring starter project → Add all the dependencies → Create three packages named as (**com.example.user.controller, com.example.user.entity, com.example.user.repository**) & create two classes named as **UserController.java, User.java** file & one interface named as **UserRepository.java** file as shown below.

Step 2: [illegible]

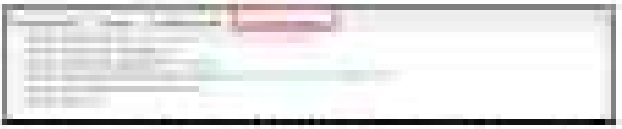
Step 3: [illegible]



Step 6: Now write code in **UserRepository.java** file as shown below:

Step 7: Now go to the **src/main/resources** folder and open the **application.properties** file and add the following code:

Step 8: Now add code in **application.properties** file as shown below:



Step 9: Now in **src/main/java**, right-click on the **User/application.java** file and go to **Run as > Java Application**.

Now the server is running successfully.

Step 8: Create a new database by running the command as 'use **database**' as shown below.

Step 9: Login to the Postman & select 'GET' method ; enter the command as localhost:8080 as shown below ; Send ; Send.

Step: Then check the response status here. If the status is 200 OK means the response has been accepted by the POSTMAN.

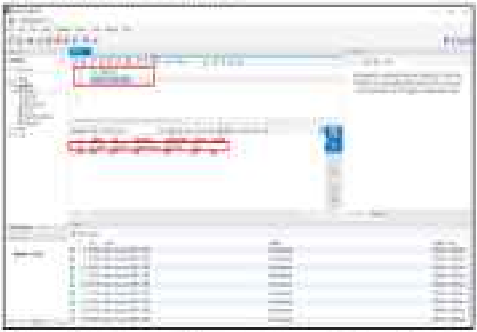
Step 69: Now write the code as shown below.

Note: To run the code, select each code line separately & press **ctrl+shift+enter** keys to view the output.

Step 71: Select 'POST' method → enter the command [illegible] → select Body → text type: JSON → enter the below code → Select → Send.

Step 12: Now go to MySQL Workbench & enter the below command in script editor. To run the code select each code line separately & press **ctrl+shift+enter** key to view the output.

Note: Use the FIRST method to restart successfully.

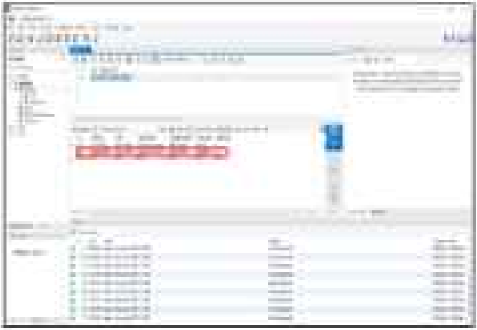


Step 13: Select "POST" method → enter the connection "localhost:9090/users" as shown below → select **Body** → raw type "JSON" → enter the below code → Send → Send.

Step 11: Navigate to MySQL Workbench & run the below command as shown below. To run the code, select each code line separately & press **ctrl+shift+enter** key to view the output.

Note: Here the POST method is created successfully.

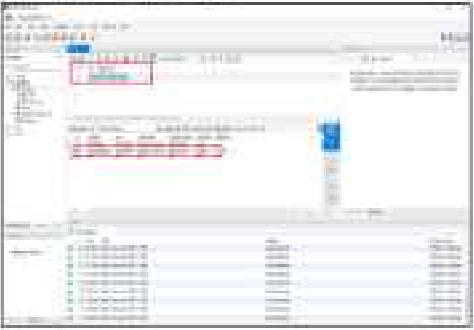
Note: Here 1's created has POST method with different id to view the PUT method clearly.



Step 12: Select **PUT** method → enter the command path as specified as "localhost:8080/...." → ... → select **Body** → text type → JSON → enter the below code → Send it → Send.

Step 16: Now go to MySQL Workbench & enter the below command as shown below. To run the code, select each code line separately & press **ctrl+shift+enter** key to view the output.

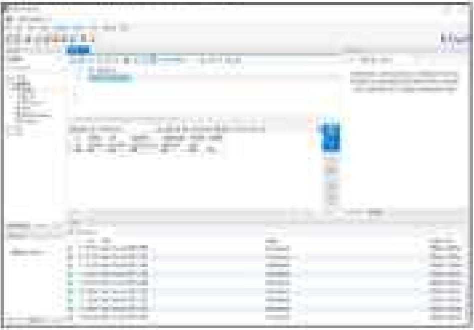
Note: Here the PUT method is updated successfully.



Step 17: Select '**DELETE**' method → enter the command which is specified as 'localhost:8080/...' as shown below → select **Body** → text type – **JSON** → enter the below code → Save it → Send

Step 08: Now go to SQL/PLSQL. We will need to enter the below command as shown below. To run the code, select each code line separately & press **ctrl + shift + enter** key to view the output.

Note: Here the DELETE is used to be deleted successfully.



WEEK – 9

1. Build user authentication flow and authorization using SpringSecurity.

Step 1: Open Eclipse IDE. Go to Help → Eclipse Marketplace. Search for Spring Tools 4 (Spring Boot Tools). Click Install and restart Eclipse. After restart, go to File → New → Other, search for Spring Starter Project.

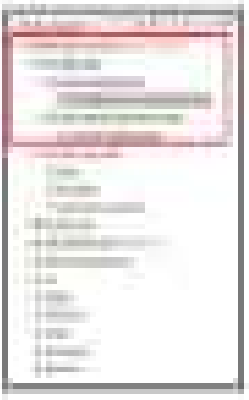
Step 2: Enter project details: Name: SpringSecurityDemo, Type: Maven, Packaging: Jar, Java Version: 17 (or 11). Click Next.

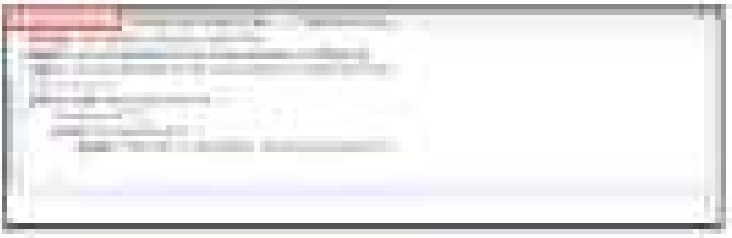
Step 3: Select the following dependencies: Spring Web, Spring Security, Thymeleaf, Spring Data JPA, H2 Database. Click Finish.

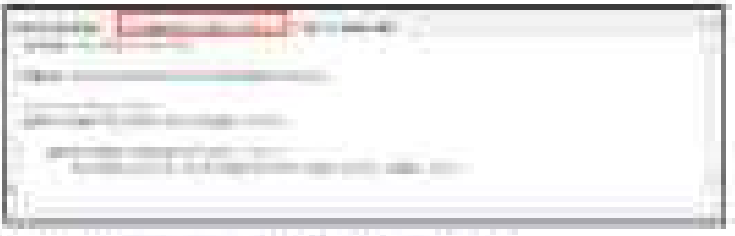
Step 6: Now add the required dependencies in the pom.xml file. Click on Finish.

Step 7: Create a new package named com.example.controller. Right click on it.

Step 8: Select new > class and name it as HomeController. Click on Finish.





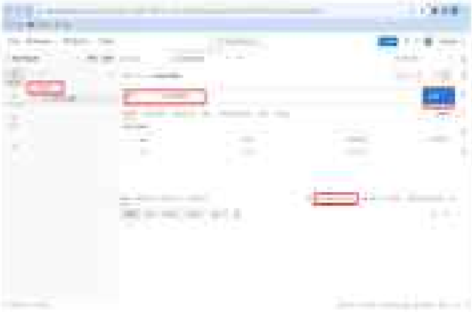


Step 11: Start the server as shown below.

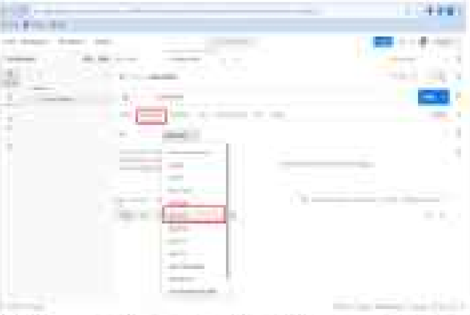
Step 12: Now login to the Postman & create a new collection as shown below.

Step 13: Select [illegible] then click on [illegible]. The [illegible]

[illegible]



Step 14: Now go to 'Authorization' tab , select the Option 'Basic Auth'.

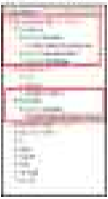


Step 15: Now enter Username & Password as mentioned below. Then Save it & click on Send to view the output.

- **<u>Implementing Code for the above Security API</u>**

Step 16: Follow the same procedure as done from step 1

Now Create the packages & classes as shown below, then add code as mentioned from step 17



Step 17: Now add code in "SpringBootSecurityApplication.java" as shown below

Step 18: Right click on the "SpringBootSecurityApplication.java" file... Run As... Spring Boot App...

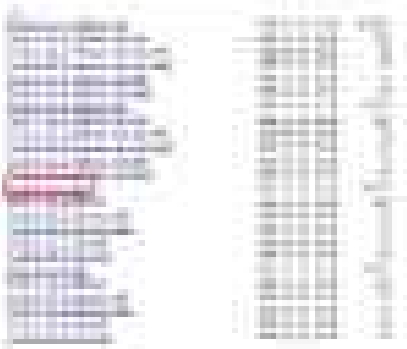
2. Writing Junit test cases for CRUD operations

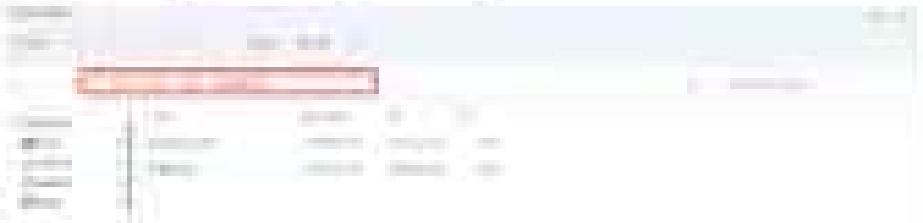
Step 1: Go to Google and search for "junit 4", select the first link shown below.

Step 2: Click on "Download and install" option.

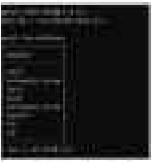
Step 3: Now click on "junit.jar" file here.

Step 4: Then select the "junit-4.13.1.jar" version to download it.

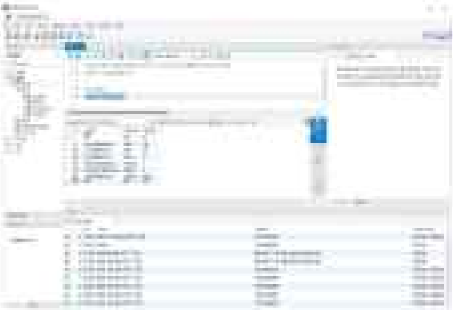




Step 11: Now create a new database in MySQL Command Line Client



Step 12: Go to MySQL Workbench and create the user table [illegible] CRUD options.

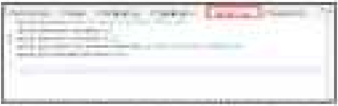


Step 13: Follow the same procedure followed for CRUD operations and add code in the classes created.

Step 16: Now add code to "[illegible]" file.

Step 17: Now add code to "[illegible]" file where main method class present here.

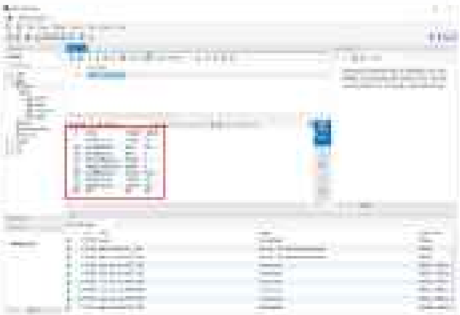
Step 18: Now add code in 'application.properties' file here.



Step 19: Go to [illegible] folder [illegible] and add code [illegible] 'CrudApplicationTests.java' file as shown here.

Step 20: Now right click on the 'CrudApplicationTests.java' file → Run As → JUnit Test

Step 21: The above test condition is successfully running in MySQL Workbench as shown below

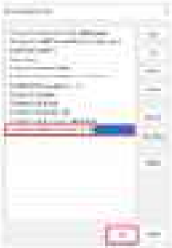


1. **Installation of Mongodb, mongosh, compass**

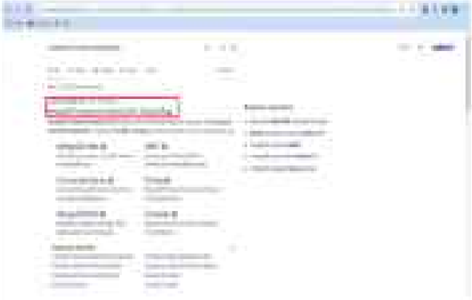
Step 1: Go to web browser ; Search for "mongodb download" ; click on the link as shown below

Step2: Click on Download link

Step 2: [illegible] for the downloaded file.



Step 3: [illegible], search for [illegible] [illegible] Compass.



Step 6: Enable the checkbox & click on 'Next'.

Step 7: Click on 'Complete' button

Step 8: Last click on 'Yes' button.

Step 8: Click on 'Next'.

Step 9: Select Skip to Install.

Step 10: Click on 'Install'

Step 12: Click next the installation is complete

Step 13: Mongo DB Compass is ready to launch.

Step 14: Then click on Visual Stu...

Step 15: Once all installation done, To verify mongod, Open cmd prompt - run the mongod --version and mongosh --version

- Use mongod [illegible]
- [illegible] mongosh [illegible] zip file [illegible]
- Use MongoDB Compass [illegible] application [illegible]

Step 16: Here to search bar [illegible] → click and open [illegible]

[illegible]

Remember: In MongoDB, a database is not actually created until it gets content

Step 17: Now search for MongoDB Compass [illegible]. Now click on "Connect" to connect with the localhost here.

Step 9: Select the database created & click on 'RESTORE' to check the data.

Week-13

1. **Create and Drop database**

MongoDB "use Database_Name" is used to create database. The command will create a new database if it doesn't exist, otherwise it will return the existing database.

Syntax:

use Database_Name

Example: If you want to use a database with name <mydb>,

use mydb

- To check your currently selected database, use the command db

db

- Your created database (mydb) is not present in list. To display database, you need to insert at least one document into it.

db.data.insert({"name":"MongoDB"}) ;show

dbs

- The **dropDatabase** command is used to drop a database. It also deletes the associated data files. It operates on the current database.

Syntax:

db.dropDatabase()

3. Create and Drop Collection

MongoDB db.createCollection(name, options) is used to create collection.

Syntax:

```
db.createCollection(name, options)
```

In MongoDB, you don't need to create collection. MongoDB creates collection automatically, when you insert some document.

MongoDB's **db.collection.drop()** is used to drop a collection from the database.

Syntax:

```
db.COLLECTION_NAME.drop()
```

drop() method will return true, if the selected collection is dropped successfully, otherwise it will return false.

1. **CRUD Operations in MongoDB**

Step 1: Open Mongosh which will directly connect to your default localhost port address → Type **show dbs** → Create a new database with name [illegible] & create a new collection with name "Student" → Now give the command "show dbs" to check the created database in present list.

- **MongoDB insert documents**

In MongoDB, the **db.collection.insert()** method is used to add or insert new documents into a collection in your database.

Insert Command to insert one document in **MongoDB & ESD**

To read the document enter the command as **db.Student.find()** & to read the document in formatted way enter the command as **db.Student.find().pretty()**

- MongoDB cannot read multiple documents. If you want to insert multiple documents to a collection, you have to pass an array of documents to the `db.collection.insertMany()` method.

- **Create an array of documents**

Define a variable named `documentsToInsert` containing an array of documents to insert.

- **Insert the documents**

Pass the `documentsToInsert` to the `db.collection.insertMany()` method.

```
db.products.insertMany(documentsToInsert);
```

To check whether the documents were successfully inserted or not, run the command `db.products.find()`.

- **MongoDB update document:**

In MongoDB, update() method is used to update or modify the existing document of a collection.

SYNTAX:

db.Collection_Name.**update**(SELECTION_CRITERIA, UPDATED_DATA)

To check whether the document is successfully updated or not, use the command db.find().pretty().

- **MongoDB Delete document:**

In MongoDB, we use **db.collection.remove()** method to remove documents from a collection. The remove() method accepts some parameters:

- JustOne: It removes only one document when set to true or 1.

Syntax:

db.collection_name.remove (DELETION_CRITERIA)

- Remove all documents: If you want to remove all documents from a collection, pass an empty query document {} to the remove() method.

Syntax:

db.collection.remove({})

The findOne() method: Apart from the find() method, there is **findOne()** method, that returns only one document.

Syntax:

db.COLLECTION_NAME.findOne()

MongoDB - Sort Records

To sort documents in MongoDB, you need to use **sort()** method. The method accepts a document containing a list of fields along with their sorting order. To specify sorting order 1 and -1 are used. 1 is used for ascending order while -1 is used for descending order.

Syntax: The basic syntax of sort() method is as follows:

db.Collection_Name.find().count(JSON)

MongoDB: Limit() method

To limit the records in MongoDB, you need to use **limit()** method. The method accepts one number type argument, which is the number of documents that you want to be displayed.

Syntax: The basic syntax of limit() method is as follows −

db.Collection_Name.find().limit(JSON)

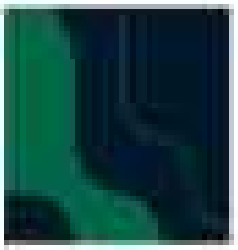
MongoDB: Skip() Method

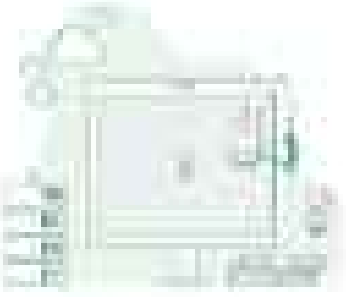
Apart from limit() method, there is one more method skip() which also accepts number type argument and is used to skip the number of documents.

Syntax: The basic syntax of skip() method is as follows −

db.Collection_Name.find(JSON).limit(JSON).skip(JSON)

4. How to Use Mongo DB on Cloud

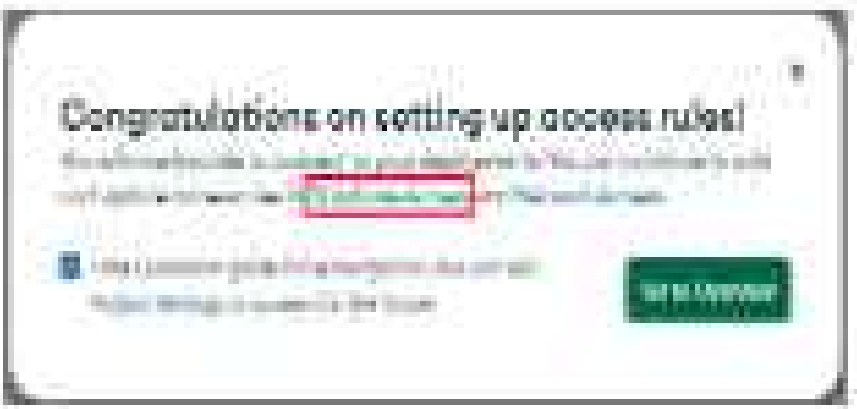




Step 5: Now select 'M0' free cluster & also select 'aws'. Then click on 'Create'.

Step 6: Enter the username & password & then click on 'Create User'.

Step 7: Now click on 'Add My Current IP Address' & click on 'Finish and Close'.


Congratulations on setting up access rules!

Step 8: Now you can edit or delete the created username and password here.

Step 9: After editing your username and password, click on 'Update User' here.

Step 10: Select 'Database' & click on 'Connect' option here.


Connect to Cluster
Connecting with MongoDB Compass
I. Choose your version of Compass
II. Copy the connection string, then open MongoDB Compass
Done


New Connection

Step 13. Then give the password & click on "Connect".

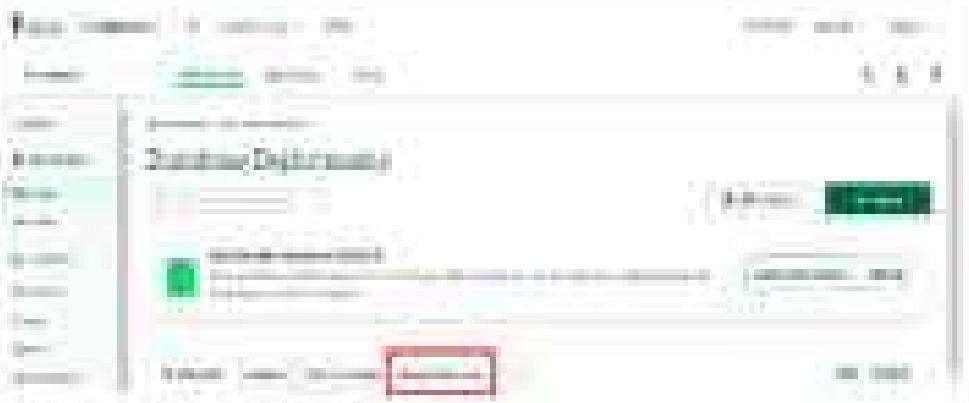
Step 14: Click on "+" [illegible]

Step 15: Create a new database with the name college & collection with the name [illegible]. And click on "Create Database" [illegible]

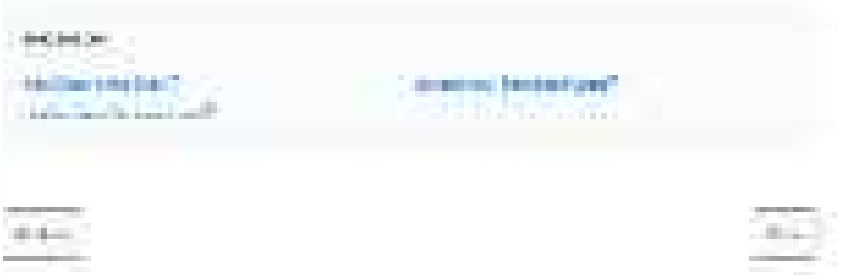

Create Database

Step 11: Your inserted document is being displayed here.

Step 12: Click on "Browse Collections" Item.

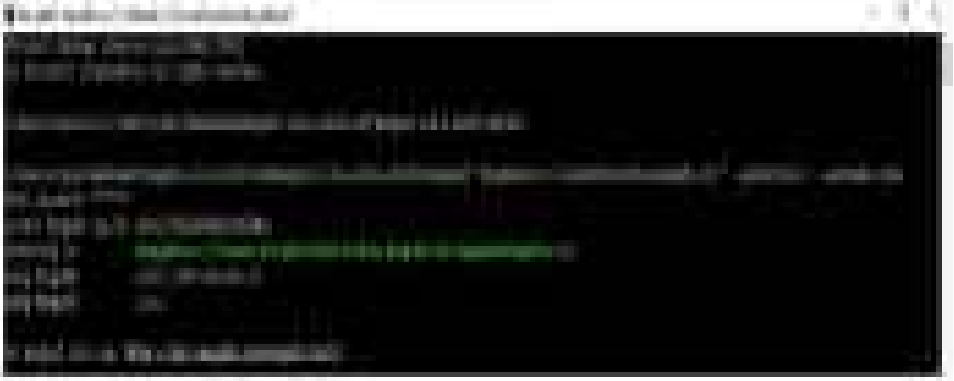


Connect to Cluster



Step 2: Now copy the link location of MongoDB Shell from the cluster page and paste it in your terminal.

Step 3: Now open the path where you copied the mongo.exe file in cmd and paste the path which you copied in the Step 2 in your terminal. This will be your terminal now.

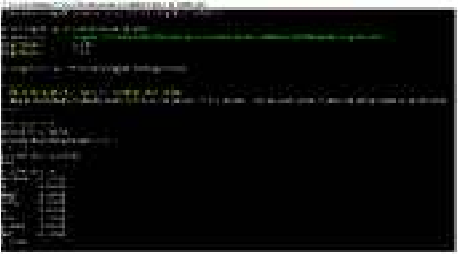


Step27: Use the existing database as shown below.

Step28: Now go to Browse collections to view the following output.

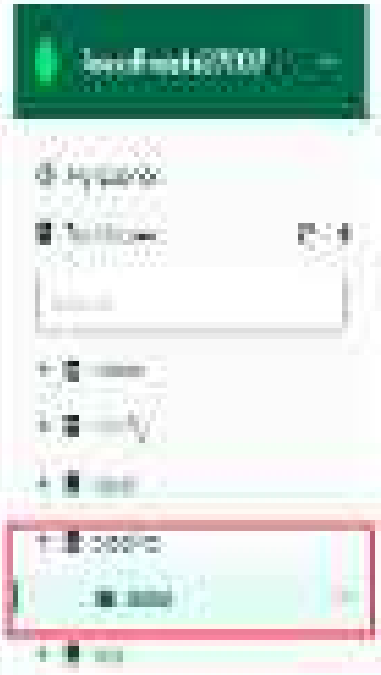
2. Aggregation concept in MongoDB.

Step1: Open Mongosh and create a new database and collection as shown below.



Step2. Navigate to [illegible] Copilot & click on "Create" option here.

Step3. Select the model database and deploy the data.




Insert Document


Insert Document

Step 11: Execution 'final' aggregate operation icon

Step 12: Navigate to pipeline tasks. The first result will be the aggregate stage.

4. Perform CRUD Operations on MongoDB through REST API using Spring Boot Starter Data MongoDB.

Step 1: Open the Eclipse IDE. Go to Categories Editions ... View ... Other ...

Step 2: Select "Spring Starter project" & Click on "Next".

Step5: Select the option as shown below & click on Next.

Step 6. [illegible] & click on 'Finish'.

Step 7. [illegible] & click on 'Finish'.

Step 6: Now create a new package named "com.example.mongodb.repository" & click on "Finish".

Step 7: Now create a new package named "com.example.mongodb.controller" & click on "Finish".

Step 8: Now create a new class named "MongodbApplication" in the package "com.example.mongodb" & click on "Finish".

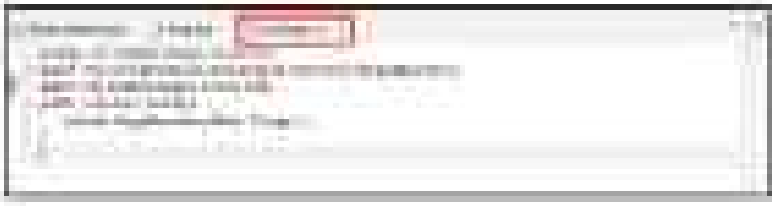
Step 9: Now create a new interface named 'BookRepo' for the package named [illegible]

Step 10. Now create a new class named 'BookController' for the package named, 'com.example.mongo1.controller' & click on 'Finish'.

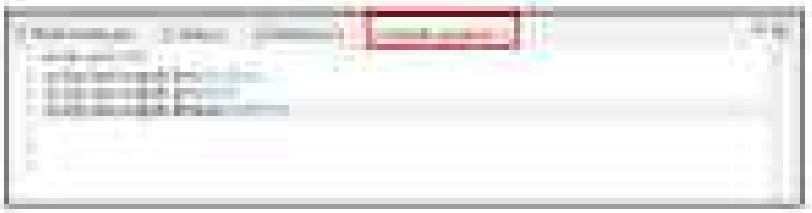
Step 11. Now add code to 'Book.java' file.

Step 12. Now add code to 'BookController.java' file.

Step 12: Save and exit the `BookShop.java` file.



Step 13: Save and exit the `application.properties` file.



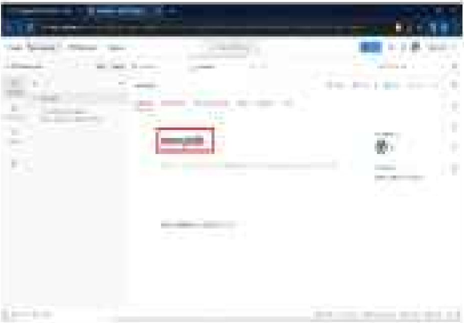
Step 14: Go to MongoDB Compass & connect to the existing URI path & then click on `Connect` button.

Step 16: Now create a new database named 'bloodbank' and collection named 'donor' in Mongodb as shown below

Step 17: Now run the main method file to do Crud as shown below.

Step 9: Observe the file server logs

Step 10: Now login to your MongoDB workspace and create a new collection named "[illegible]" as shown below:



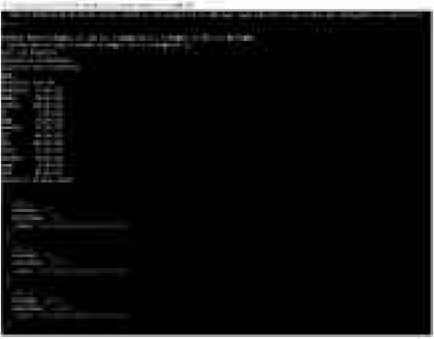
Step 20: Perform GET method by entering the command as **localhost:9090/book/2/Books** here.

Step 21: Perform POST method by entering the command as **localhost:9090/addBook** here.

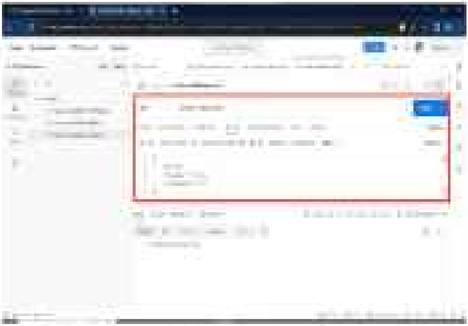
Step 22. Perform POST method to insert one document as shown below.

Step 23. Perform POST method to another document as shown below.

Step 26: Go to [illegible] and the [illegible] as shown below



Step 27: Perform PUT method by entering the command as **localhost:3000/update/**[illegible]



Step 20: Use the below command to check whether update operation is reflecting in Mongodb.

Step 21: Perform DELETE method by entering the command as localhost:8000/delete/2 here

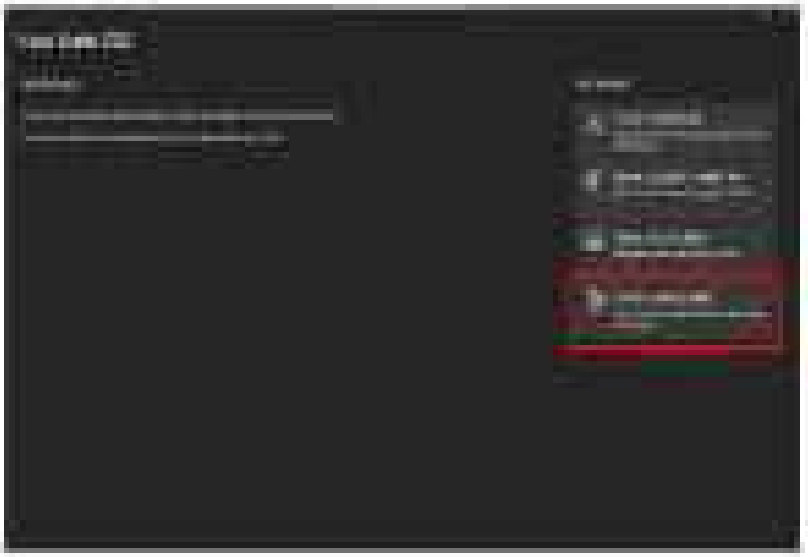
Step 05: Use the below command to check whether delete operation is working or not in MongoDB.

Step 06: Follow the same procedure to perform CRUD operations on MongoDB Compass through SYSTEMAN as well.

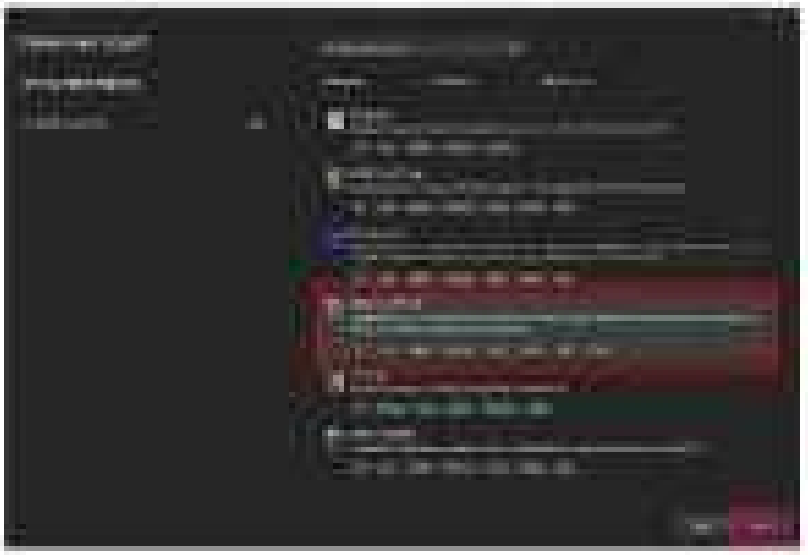
Week 13

1. Integrate the work of each group and carry out integration testing.

Step 1: Open Microsoft Visual Studio and click on Create a new Project.



Step 2: Select ASP.NET Core Web API & click on Next.



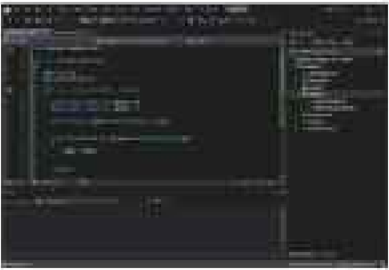
Step 3: Give Project name & click on Next.

Step5: Select Add & click on 'Controller'.

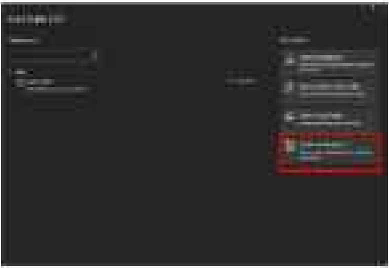
Step6: Click on Add.

Step 6: Give a name & click on Add.

Step 7: Add code in Calculator Controller.cs file



Step 10: Now close the other windows and open the Microsoft Visual Studio and now click on Create a new Project.



Step 11: Select ASP.NET CORE Web API & click on Next.

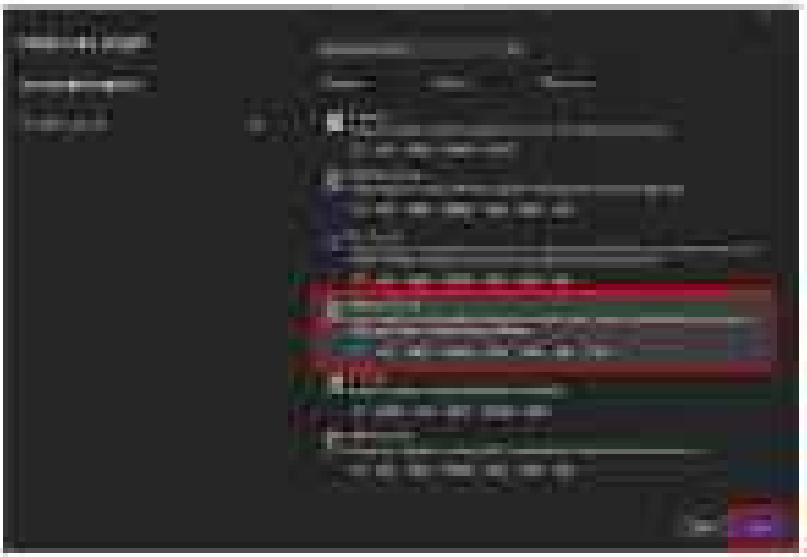


Fig 12: Clone Project window in Android Studio

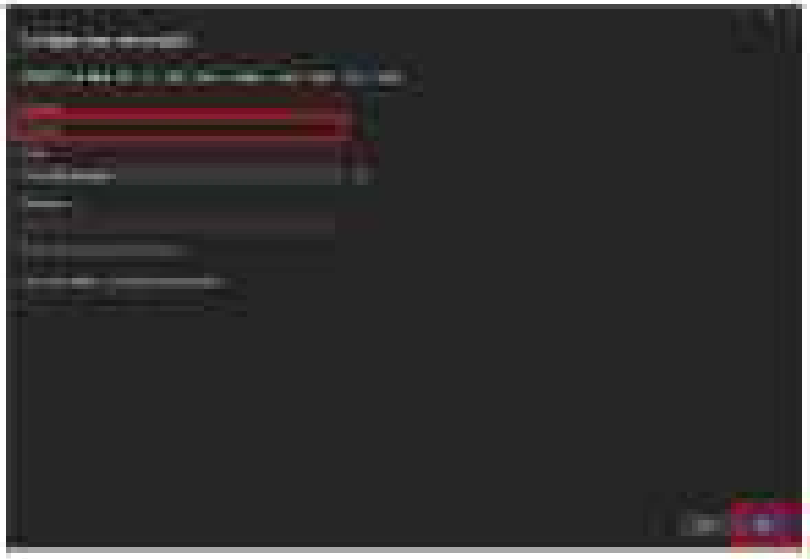
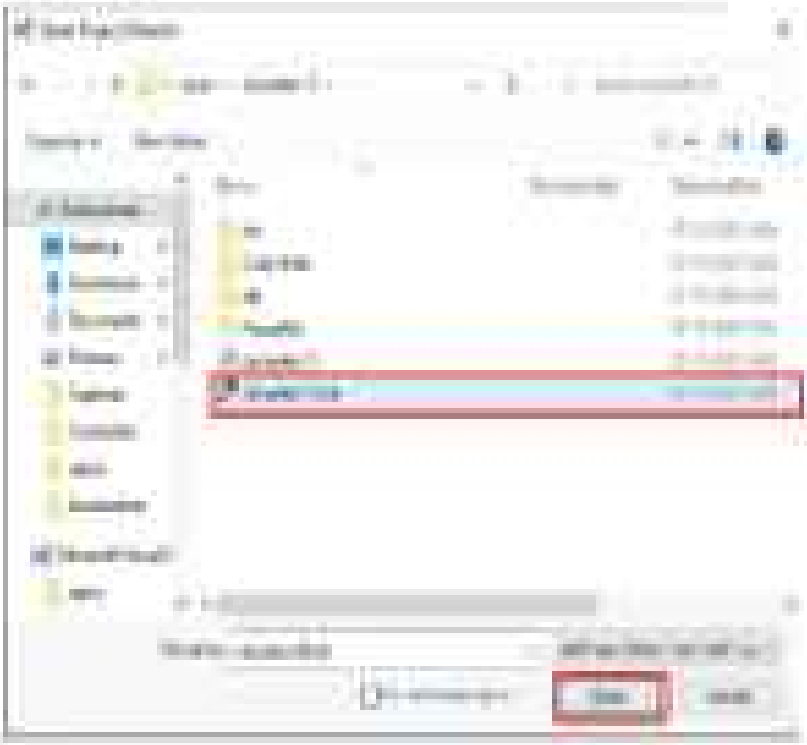


Fig 13: Gradle Sync Finished in Android Studio



Step 12: Now click on Add Members

Step 13: Select Try it out option & enter any 2 members. Then click on Execute

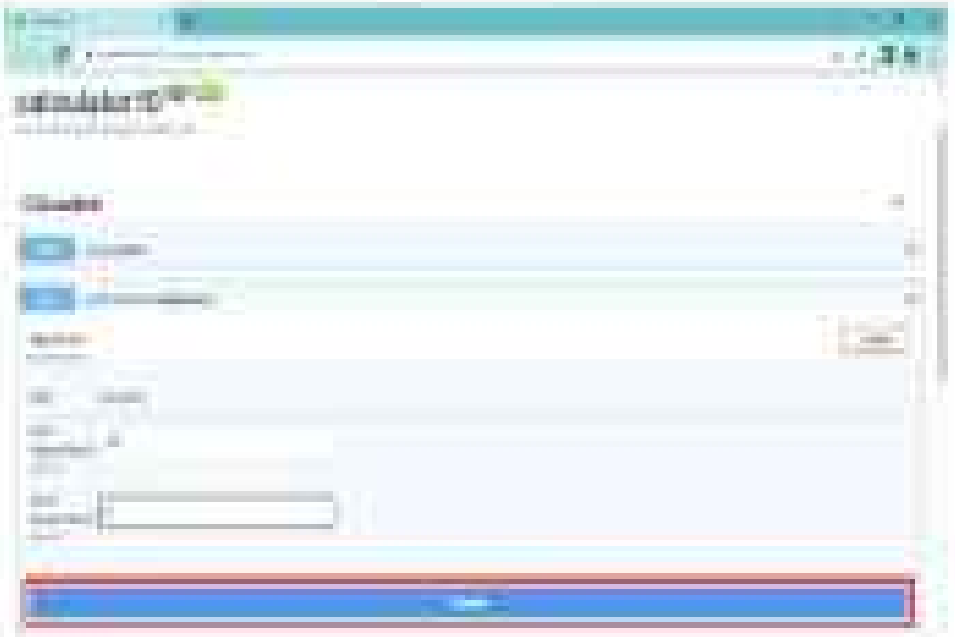
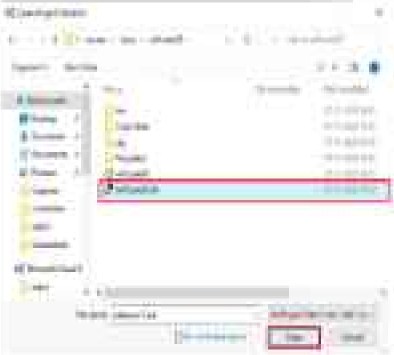


Fig 22. Choose the emulator

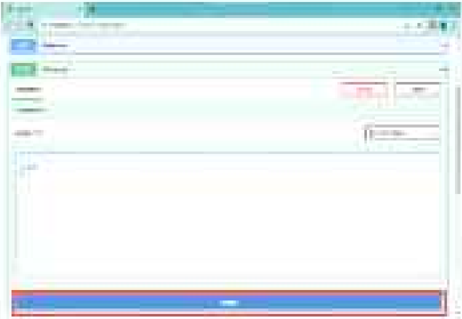
Fig 23. Code execution is successful



[illegible]

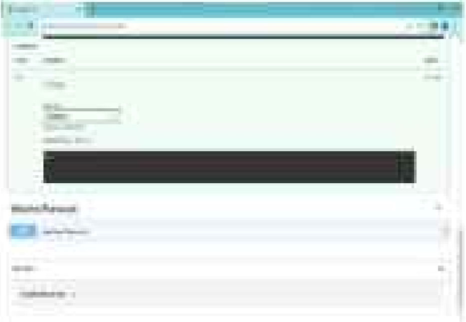
[illegible]

Step 30: Select **Try it out** option & enter any Function. Then click on **Execute**.



Step 31: Observe the result.

Step 12: Code executed successfully.



2. **Build docker image with docker file.**

Step 1: Go to Taskbar ... search for **Turn Windows features on or off** & enable the 2 checkboxes here (**Windows Hypervisor Platform, Windows Subsystem for Linux**). Then click on **OK**.

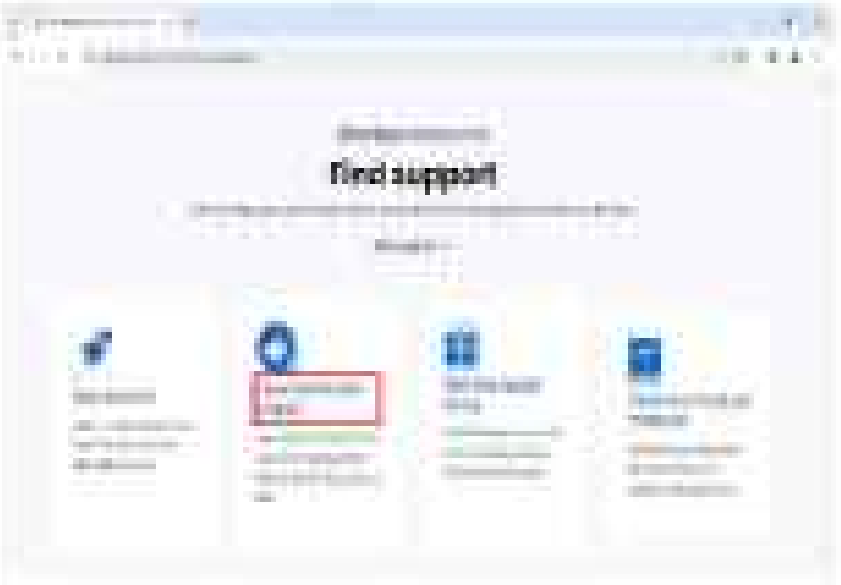
Step 2: Open Command Prompt and enter the command as **wsl --update** to check the version.

Step 3: Now restart the computer and enter **docker --version** to check the version.

Installing your Docker Desktop

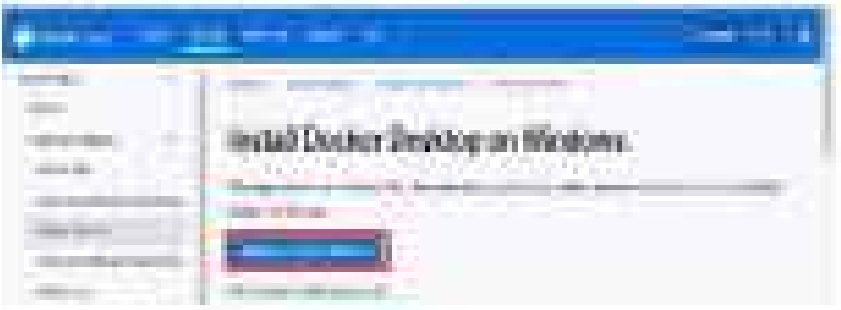
Go to Web Browser >> Search for Docker >> Click on the second link as shown

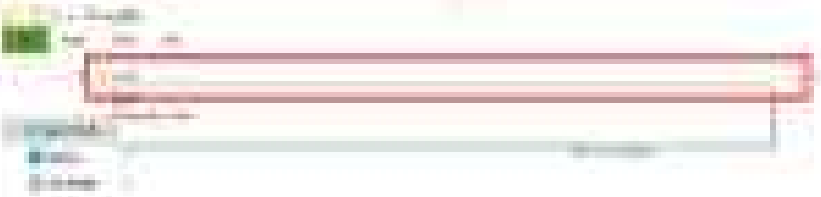
Scroll down to select **Download and install** option here.

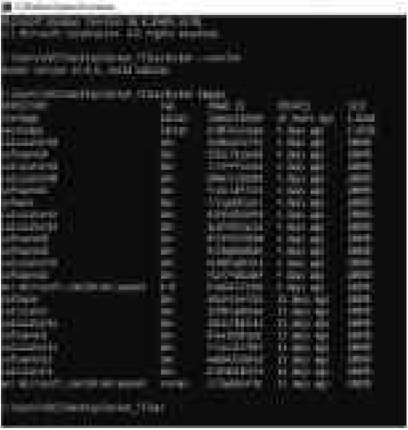


Now click on **Docker Desktop for Windows** here.


Get Docker

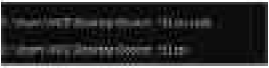

Install Docker Desktop on Windows





Step 7: Now type **python** to check the installation version of python.

Step 8: Now enter the command **code .** to directly open it in the VS Code.

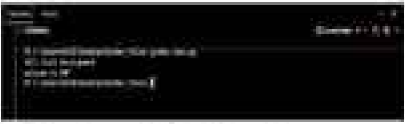


Step 9: Create 2 files named as **Dockerfile** & hello.py as shown below.

Step 12: Enter the below code in app.py file.

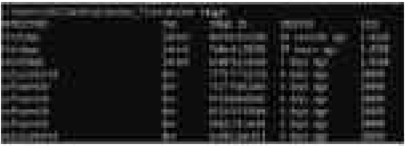
Step 13: Enter the below code in Dockerfile as well.

Step 14: Go to Terminal -> New Terminal -> enter the command as python app.py



Step 15: Now enter the command as docker build -t flaskapp .

Step 14: In the Editor [illegible] enter the command as **docker images** to view created images.



Step 15: Enter the command as **docker run --name mypythonfile -it mpy** to run the docker image.

Step 16: Now open the Docker Desktop, your created docker image is displayed there.

Follow the same procedure for creating a Docker image through Java file.

Step 1: Now create a new folder named **Docker_Java** in the Desktop.

Step 2: Go to the folder path and enter cmd. It will open the command prompt. Or directly open the VS Code. Now create a file named as **Dockerfile-A_sample.java** in the same folder.

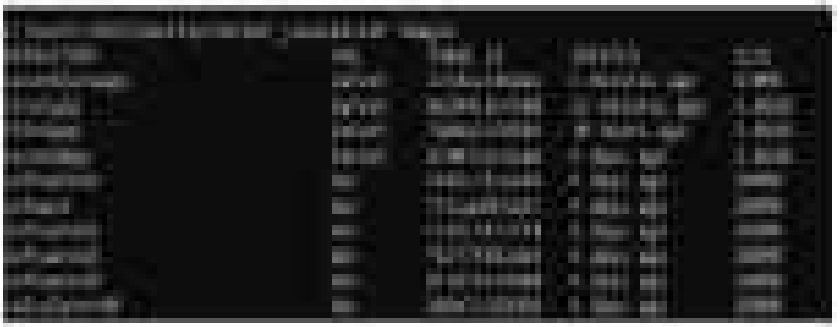
Step 3: Open the Dockerfile-A_sample.java file.

Step 4: Also create the Dockerfile named as **Dockerfile** here.

Step 5: Go to Terminal → New Terminal → Now write the command **javac sample.java** & **java sample** here

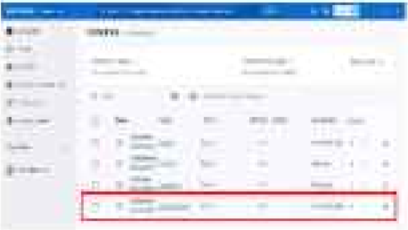
Step 11: Enter the command as **docker build -t wordpressapp .** to build a docker image

Step 12: In the folder path and, once the command as **docker images** to view the list of created docker images



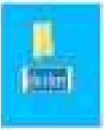
Step 13: Now enter the command as **docker run --name mylinuxapp wordpressapp** to run image

Step 25: Then you can see the image has been created in Docker Desktop.

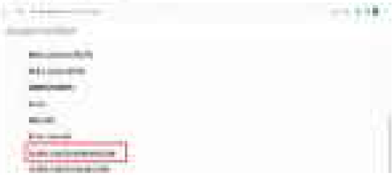


3. Create docker container from docker image & Run the docker container

Step 1: Create a new folder in desktop named as "**docker**" as shown below.

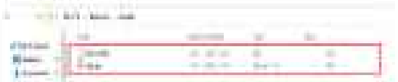


Step 2: Open the official website [illegible] in web browser and click on Creating Image for python application



Step 3: Now download the 2 files "**Dockerfile**" & "**test.py**" which are displayed here

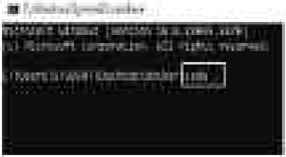
Step 4. Then copy & paste the 2 downloaded files to the new folder created as "**docker**" as shown below.



Step 5. Go to Desktop → docker → cmd as shown below.

Step 6. Now enter '**code .**' as shown below.

Note: This command will directly open the VS Code.



Step 7. Now paste the default code in "**Dockerfile**" as shown here.

Step 8: Now enter the default code in **'test.py'** as shown below.

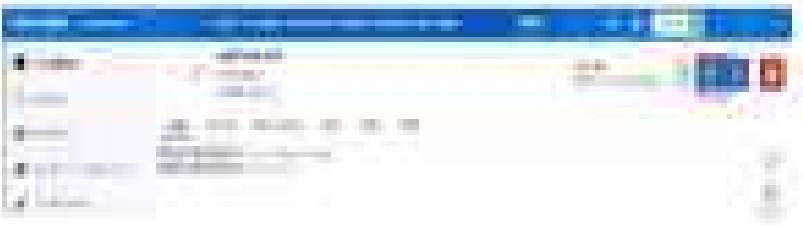
<u>Note:</u> You can change the code if you want.

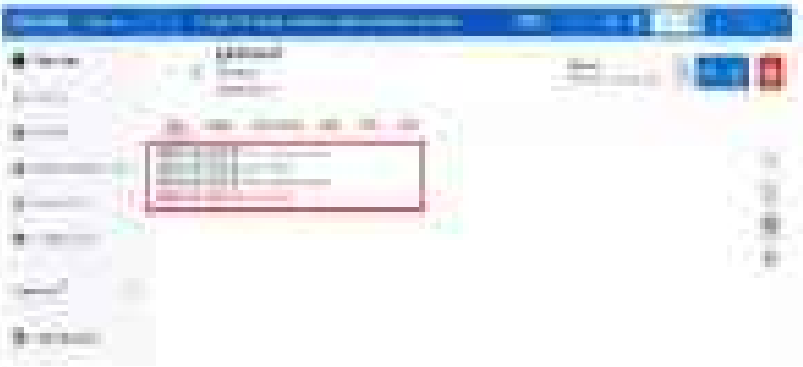
Step 9: Now open Terminal > New Terminal > Enter the command **'docker build -t pythonapp .'** to create the docker image & **'docker run --name pythoncont pythonapp'** to run the docker container with image.

<u>Note:</u> **pythonapp** - name of docker image

pythoncont - name of the docker container

Step 10: Go to Docker Desktop -> Containers -> you can see the created container with image here.

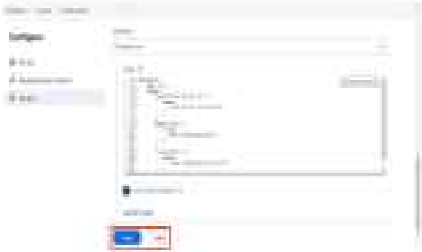




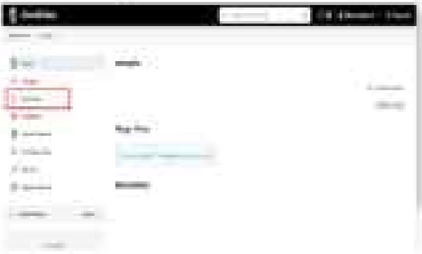
4. Jenkins Pipeline

Step 2: Enter an item name here & select 'Pipeline'. Then click on **ok**.

Step 3: Scroll down and add a sample code under Script section. Then click on **Apply & Save**.



Step 4: Now click on **Build Now** option then.

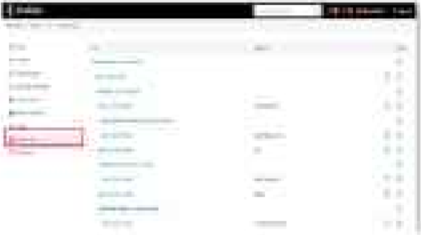


Step 5: Now observe the three stages viz. **Clone Repository, Test, Deploy** as shown below.

Step 6: Then click on **Console Output** then.

Step 7: Now observe the three stages running successfully.

Step 8: Then click on **Pipeline Steps** to view the pipeline steps as shown below.

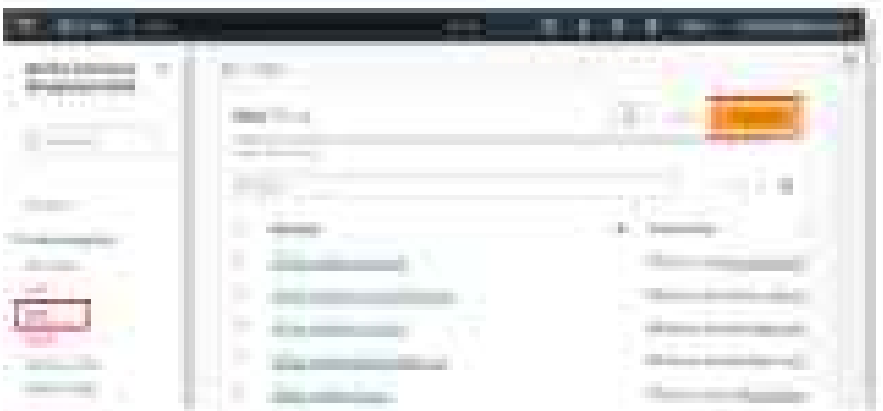


WEEK – 12

1. Kubernetes Configuration as Deployment, Cluster creation, Services & Load Balancer.

Step 1: Login to your AWS account & search for "EKS" (Elastic Kubernetes

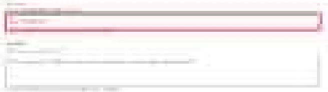
Step 2: Now go to Roles : click on Create Role button



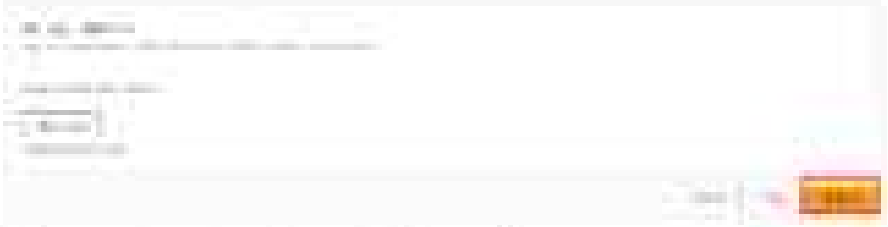
Step 3: Select EKS-Cluster and click on Next.

Step 4: Click on **Next**.

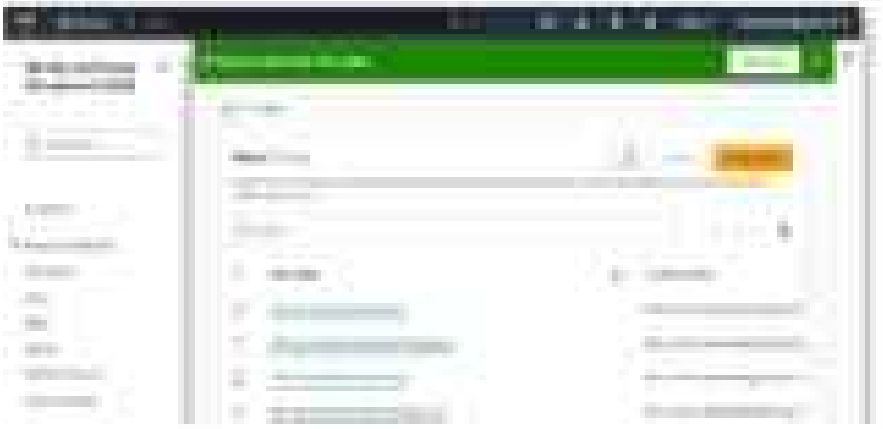
Step 5: [illegible] as shown here.



Step 6: Scroll down and click on Create Role.

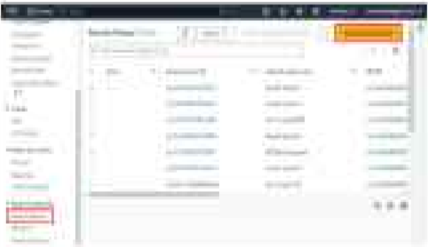


Step 7: Now you can see the role is successfully created here.

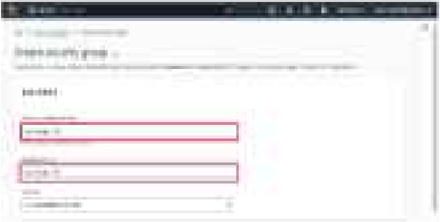


Step 8: Search for EC2 and click on Role Link here.

Step 9: Go to Security Groups & click on Create security group here.



Step 10: Give the name for Security group here. Then copy the same name & paste to the Description as shown here.

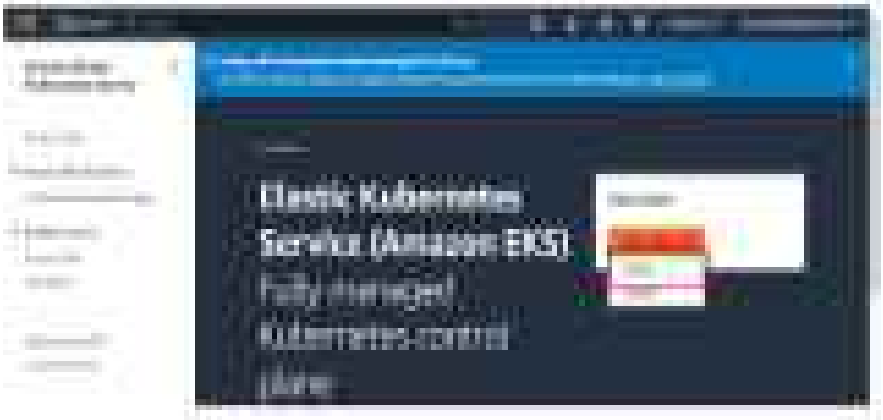


Step 11: Now add the rules as shown below & click on **Create Security Group** here.

Step 12. [illegible]

Step 13. [illegible]

Step 14: Now select **Create** option here.



Step 15: Give the name as **eks-cluster-01** , select the Kubernetes version as **1.25** , select the service role as **eksClusterRole** which we created in the previous step.

Step 11. Select **Public** and then click **Next** button.

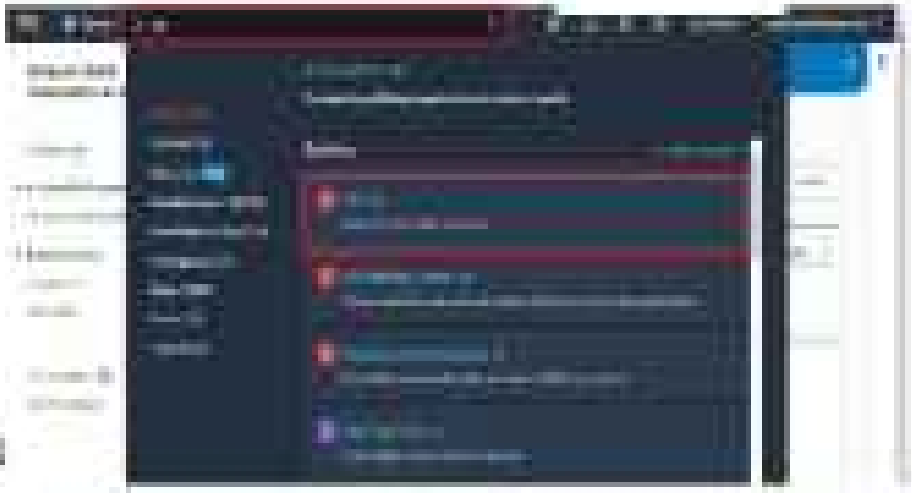
Step 12. Click **Next** button.

Step 18: Then click on Next button.

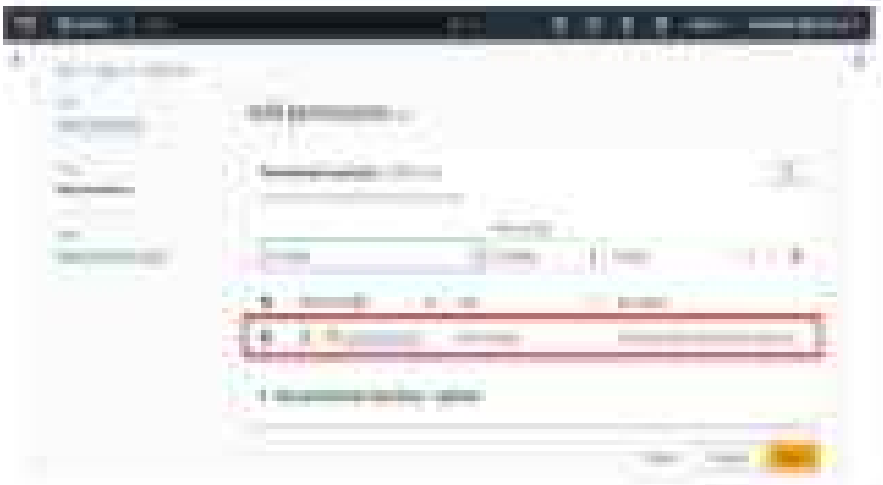
Step 19: Again click on Next button.

Fig.02: [illegible]

Fig.03: [illegible]



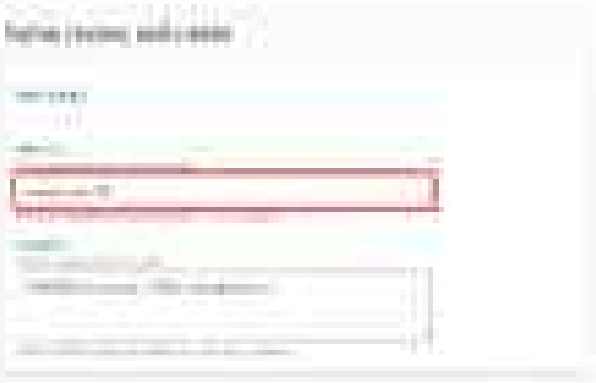
Step 26: Select the policy name as AmazonEKSWorkerNodePolicy | ec2.



Step 27: Select the policy name as AmazonEKS_CNI_Policy | ec2.

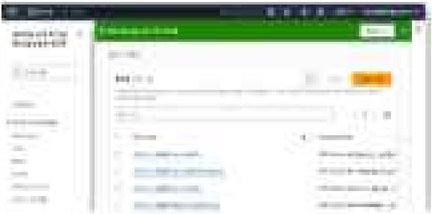
Step 28: Select the policy name as AmazonEC2ContainerRegistryReadOnly then click on Next.

Step 29: Now give the Role name as ecsInstanceRole and click on



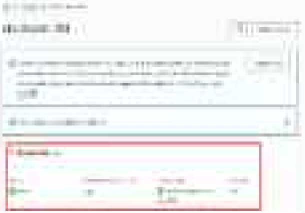
Step 9: Check that all the permissions of the policies are attached to create. Click on **Create Role**.

Step 10: Your role is successfully created.



Step 22: [illegible]

Step 23: [illegible]



Step 34: Select **Compute** & click on **Add node group** option from

Step 35: [illegible]

Step 26: Give the Scaling policy name, configure as 1 from 4. Click on Next button.

Step 27: Click on Next button.

[illegible]

[illegible]

Step 12: [illegible]

Step 13: Select **Load Balancers** [illegible]

Index: