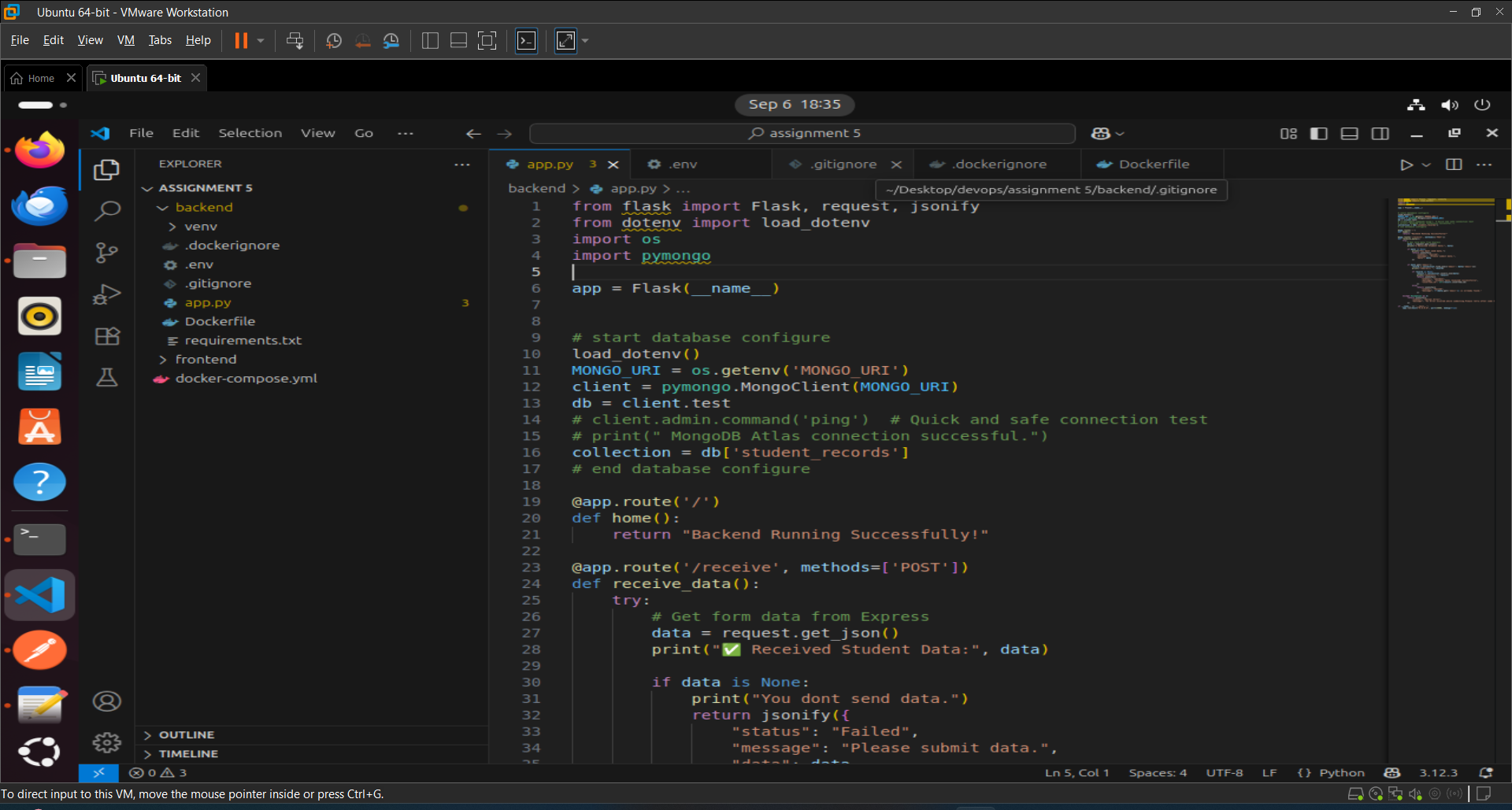
1. **Frontend (Node.js with Express)**:

* Create a frontend using Express and Node.js.
* Include a form similar to the one from the Flask Assignment 2.
* Configure the form to send a request to the Flask backend.
* **Backend (Flask)**:
* Use the Flask backend to handle the form submission and process the data.
* **Folder Structure**:
* Organize the project with separate folders for the frontend and backend.
* **Docker Configuration**:
* Create a Dockerfile for both the frontend and backend.
* Write a .yaml file (Docker Compose) to connect both services in the same network.
* Upload both images to docker hub and push your whole code to github and add the node\_modules and other non required files(.vscode) in .gitignore

Ans:

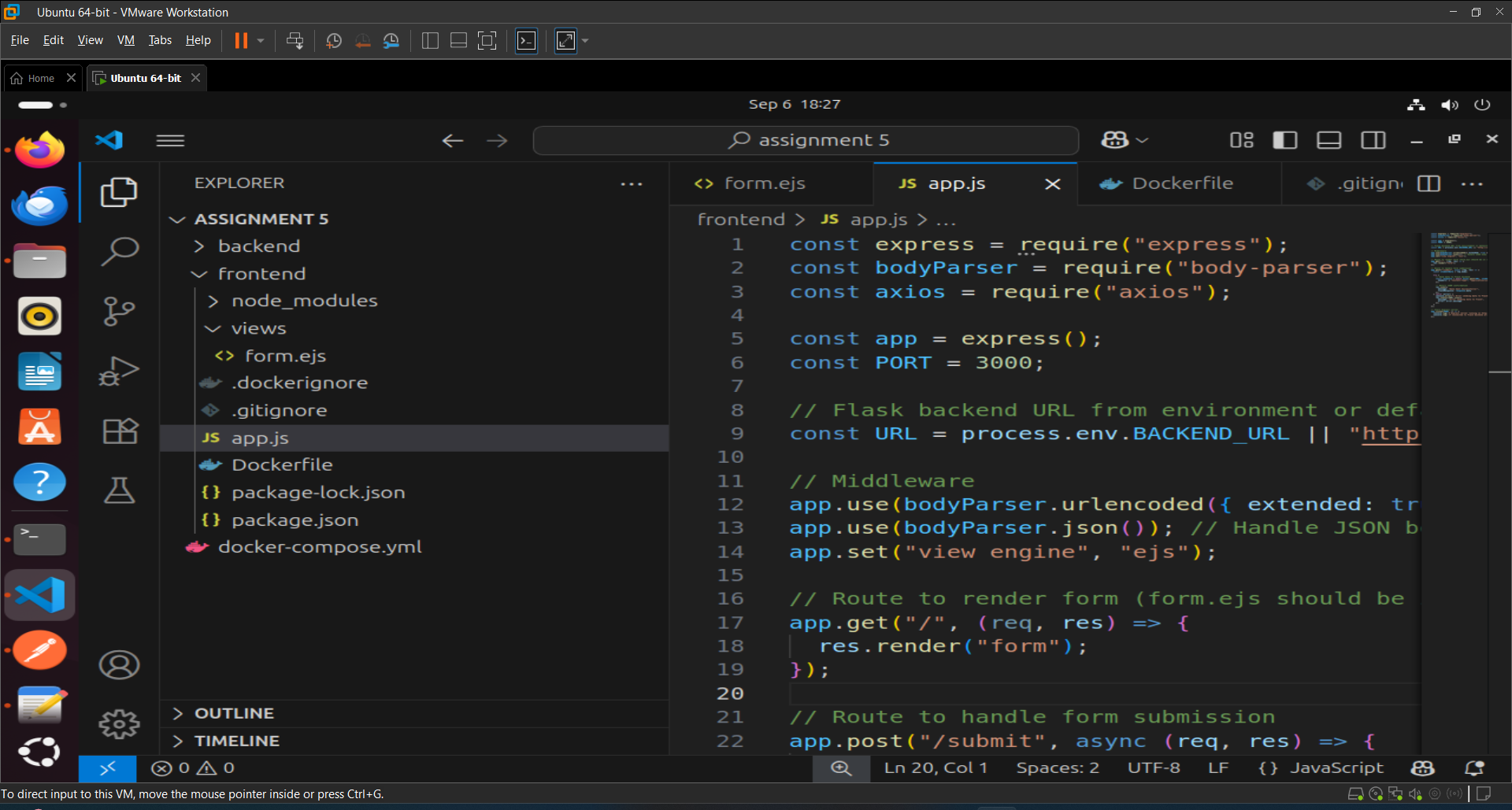
Step 1: Lets create two directory one is frontend and another is backend and start creating backend in flask.



**Backend Directory:** This direcotry contains files are:

* [**app.py**](http://app.py): This file cotains app route(“<https://localhost:5000/receive>”) with post method when the request received the form data will furture process and store at mongo database.
* .**env**: This file contains the mongo\_db url.
* .**dockerignore**: This file takes the file or directory name that need not be containerized like **venv** (becasue its a virtual environment).
* .**gitignore**: This file also contains the file or direcotry name that no need to store in our git repository such as **venv.**
* **Dockerfile**: This file is used to create docker image that will further help to build containers.

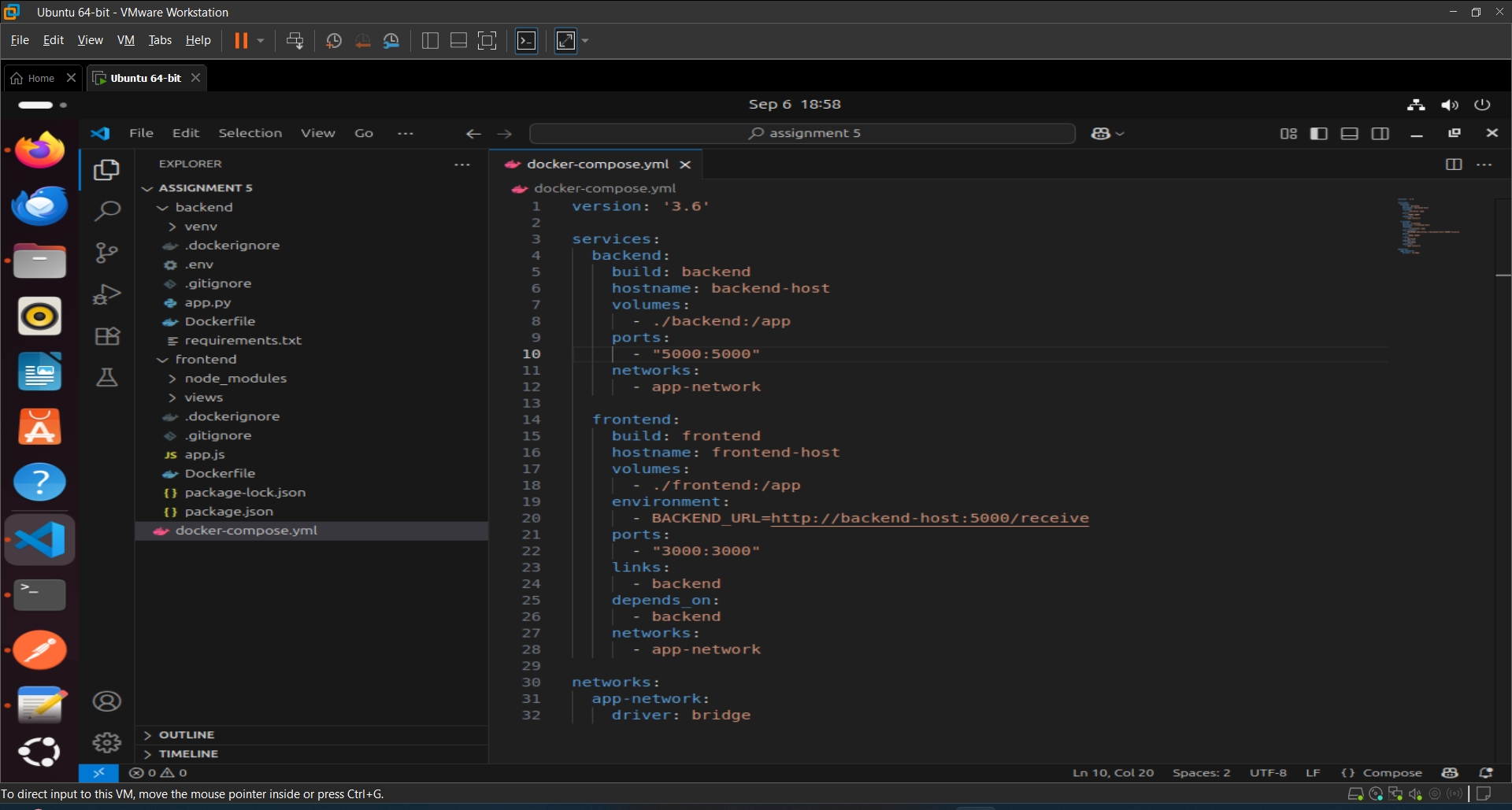
Step2: Lets create another directory named frontend and start creating frontend using Express and [Node.js](http://node.js)



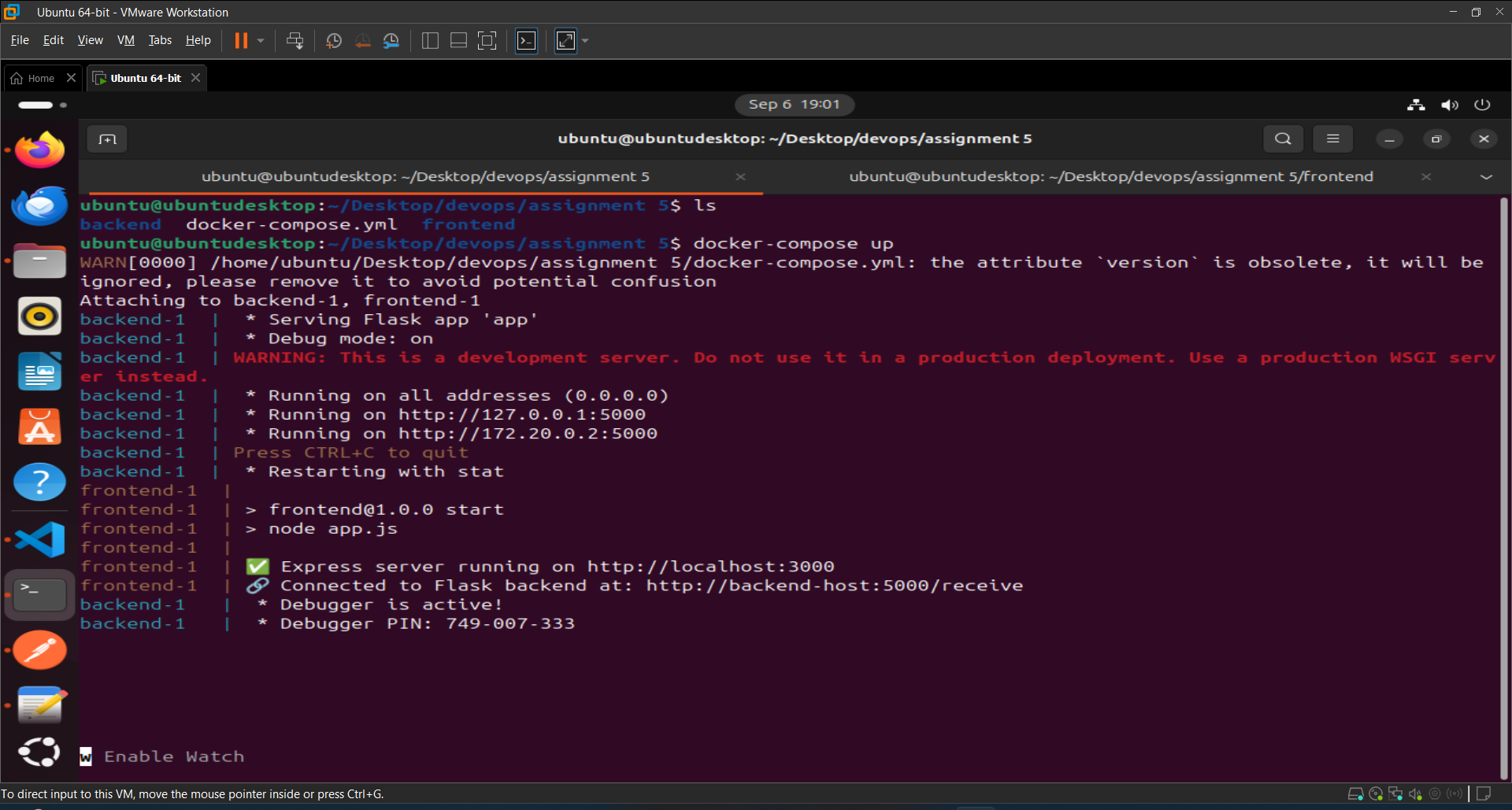
**Frontend Directory:** This direcotry contains files are:

* [**app.js**](http://app.js): This is our express app that takes request from “[form.js](http://form.js)” and sent data to our flask app.
* **views/form.ejs**: This is our html file it will render the form to the user as frontend and when user hit the **“submit”** button then the request send to express server and express server send this response to the flask app and take the response back to the **flask app** and show the response back to the frontend.
* .**dockerignore**: This file takes the file or directory name that need not be containerized like **node\_modules** (becasue its a virtual environment).
* .**gitignore**: This file also contains the file or direcotry name that no need to store in our git repository such as **node\_module**.
* **Dockerfile**: This file is used to create docker image that will further help to build containers.

Step 3: Now lets create a **“docker-compose.yml”** file to run both containers so that they can communicate to each other.



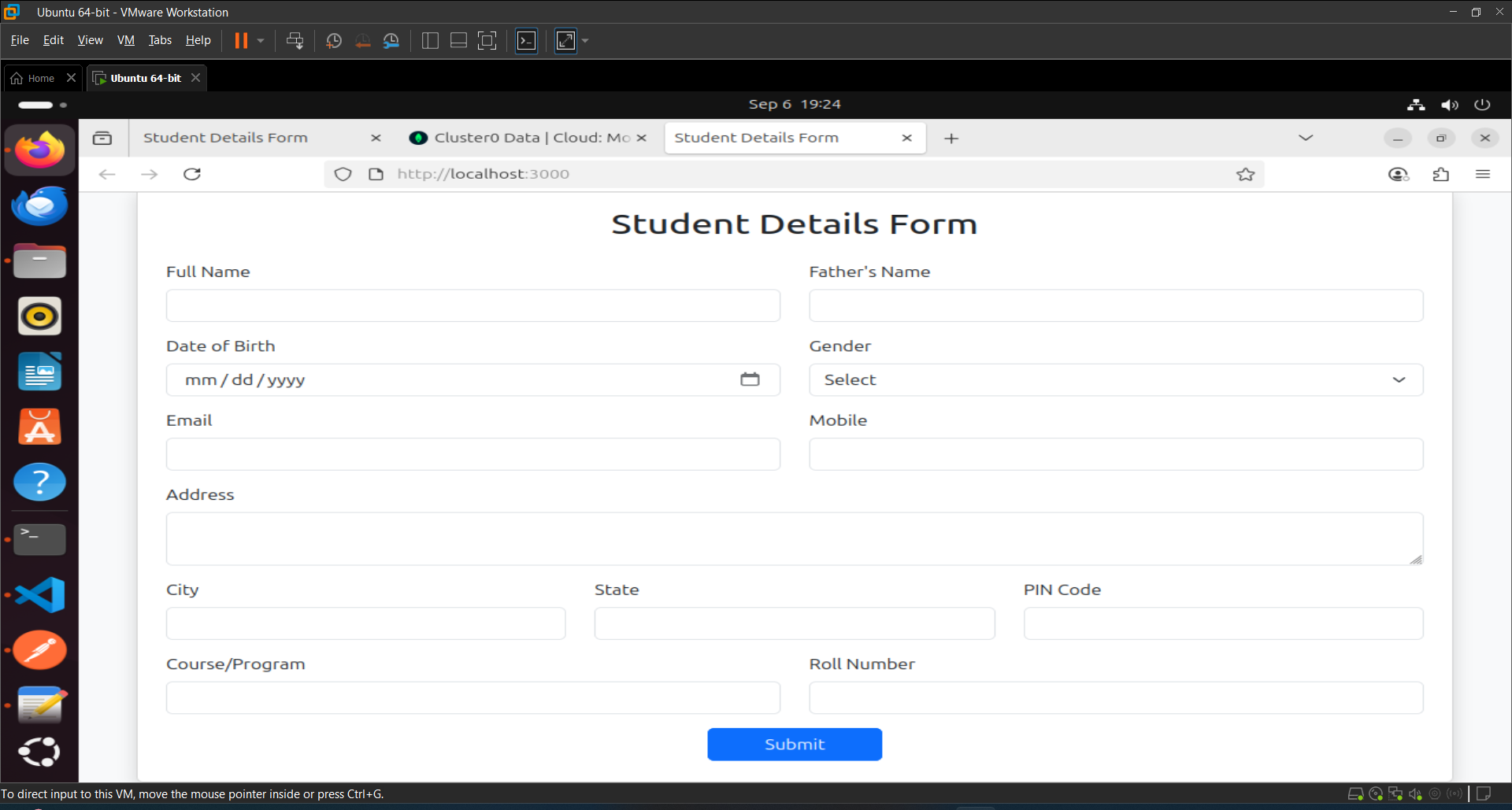
Step 5: Now run the **“docker-compose.yml”** file.



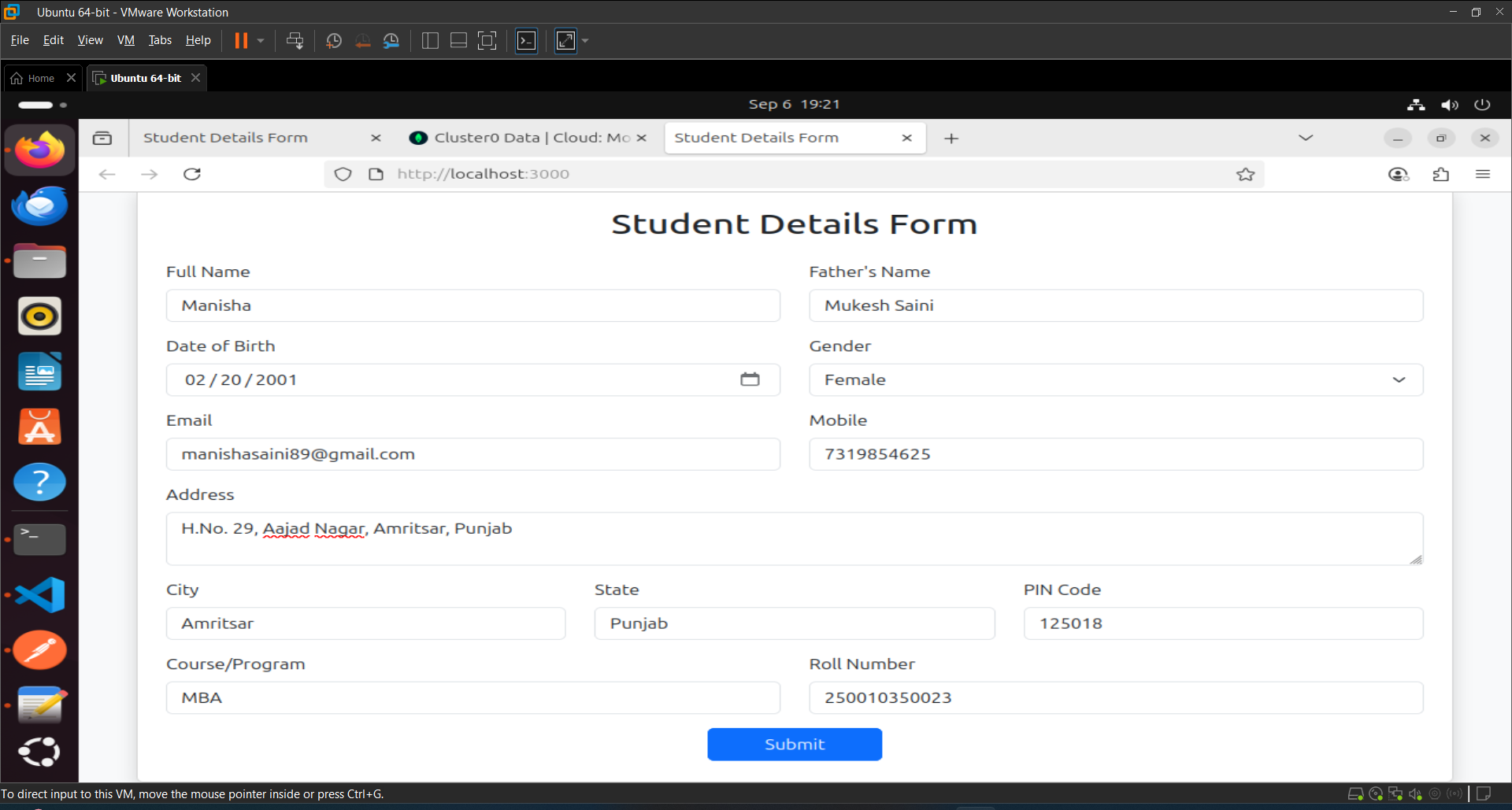
In the above image we can see our frontend and backend app are running and both have different url such as:

1. **Frontend url:** Express server running on <https://localhost:3000>
2. **Backend url:** Flaask server running on <http://127.0.0.1:5000>
3. **Connected to both services:** This url <http://backend-host:5000/receive> is used where both services are connected to each other.

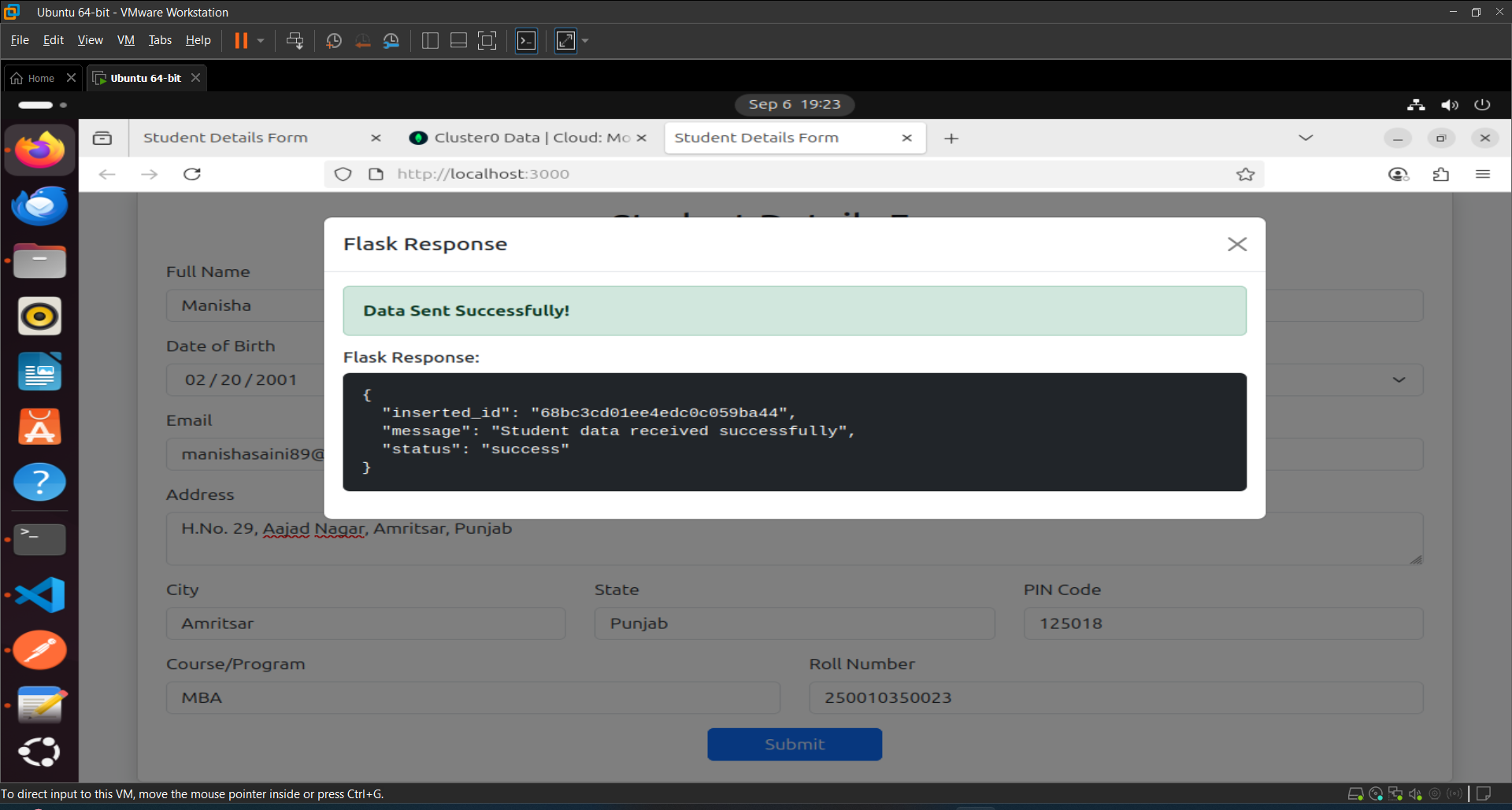
Step 6: Now visit the express app url to render UI at “<https://localhost:3000>”.



Step 7: Now fill the form and click on submit button.



Step 8: After clicking on submit button you can see the output that the data has been successfully saved to **mongo database.**



This is the screenshot of **mongo database** that data has been saved successfully.

