

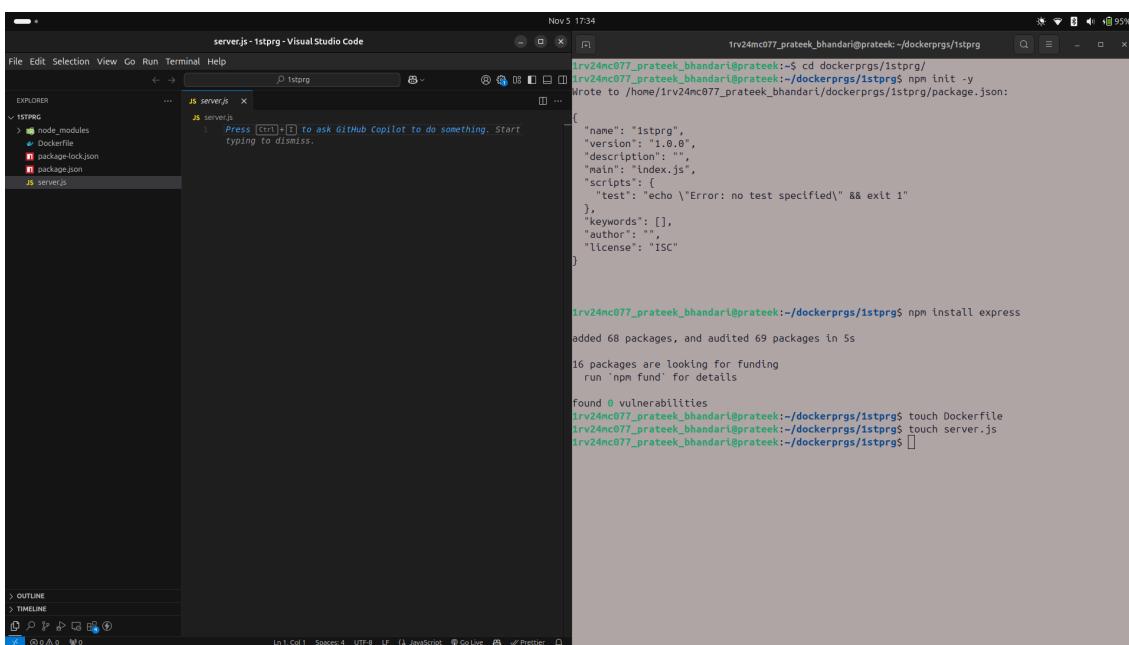
# Program 1

## Build a docker container from custom Dockerfile

step 1: Initialize node.js project

→ npm init -y

step 2 : Install dependencies



The screenshot shows a Visual Studio Code interface with a terminal window open. The terminal shows the following command sequence:

```
Nov 5 17:34
1rv24mc077_prateek_bhandari@prateek:~/dockerprgs/1stprg
1rv24mc077_prateek_bhandari@prateek:~/dockerprgs/1stprg$ npm init -y
Wrote to /home/1rv24mc077_prateek_bhandari/dockerprgs/1stprg/package.json

[ "name": "1stprg",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

1rv24mc077_prateek_bhandari@prateek:~/dockerprgs/1stprg$ npm install express
added 68 packages, and audited 69 packages in 5s
16 packages are looking for funding
  run 'npm fund' for details

found 0 vulnerabilities
1rv24mc077_prateek_bhandari@prateek:~/dockerprgs/1stprg$ touch Dockerfile
1rv24mc077_prateek_bhandari@prateek:~/dockerprgs/1stprg$ touch server.js
1rv24mc077_prateek_bhandari@prateek:~/dockerprgs/1stprg$ 
```

step 3 : Create your server file (server.js)

→ my server is listening at port 3000

The screenshot shows the Visual Studio Code interface with the title bar "server.js - 1stprg - Visual Studio Code". The menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The search bar at the top right contains the text "1stprg". The Explorer sidebar on the left shows a project structure for "1STPRG" with files: node\_modules, Dockerfile, package-lock.json, package.json, and server.js. The main editor area displays the "server.js" file with the following code:

```
const express = require('express');
const app = express();
const port = 3000;

app.get('/', (req, res) => {
  res.send('Hello my server is running successfully!');
});

app.listen(port, () => {
  console.log(`Server is running at port ${port}`);
});
```

## step 4 :Writing the Dockerfile

The screenshot shows the Visual Studio Code interface with the title bar "Dockerfile - 1stprg - Visual Studio Code". The menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The search bar at the top right contains the text "1stprg". The Explorer sidebar on the left shows a project structure for "1STPRG" with files: node\_modules, Dockerfile, package-lock.json, package.json, and server.js. The main editor area displays the "Dockerfile" with the following content:

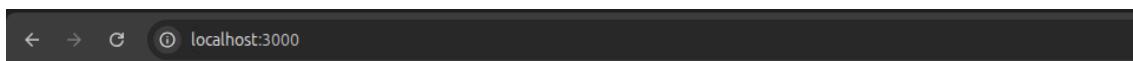
```
FROM node
LABEL author="Prateek"
LABEL description="A simple Node.js web server"
WORKDIR /app
COPY package*.json .
RUN npm install
COPY . .
ENV PORT=3000
EXPOSE 3000
CMD ["node", "server.js"]
```

## step 5 :Building Your Docker Image

## step 6 : Running Your Container

```
1rv24mc077_prateek_bhandari@prateek:~/dockerprgs/1stprg$ ls
Dockerfile  node_modules  package.json  package-lock.json  server.js
1rv24mc077_prateek_bhandari@prateek:~/dockerprgs/1stprg$ docker build -t node-app1 .
[+] Building 3.1s (10/10) FINISHED                                            docker:default
  => [internal] load build definition from Dockerfile                      0.0s
  => => transferring dockerfile: 231B                                         0.0s
  => [internal] load metadata for docker.io/library/node:latest            0.0s
  => [internal] load .dockerignore                                         0.0s
  => => transferring context: 2B                                           0.0s
  => [1/5] FROM docker.io/library/node:latest                            0.0s
  => [internal] load build context                                         0.2s
  => => transferring context: 2.70MB                                       0.1s
  => CACHED [2/5] WORKDIR /app                                           0.0s
  => [3/5] COPY package*.json ./
  => [4/5] RUN npm install                                              2.3s
  => [5/5] COPY . .
  => exporting to image                                                 0.3s
  => => exporting layers                                              0.2s
  => => writing image sha256:fb304fa7a175d705146441b4e342a10f8ec8648471bcd1d289720418af 0.0s
  => => naming to docker.io/library/node-app1                           0.0s
1rv24mc077_prateek_bhandari@prateek:~/dockerprgs/1stprg$ docker run --name application1 -p 3000:
Server is running at port 3000
```

step 7 : Open browser and visit localhost:3000



Hello my server is running successfully!