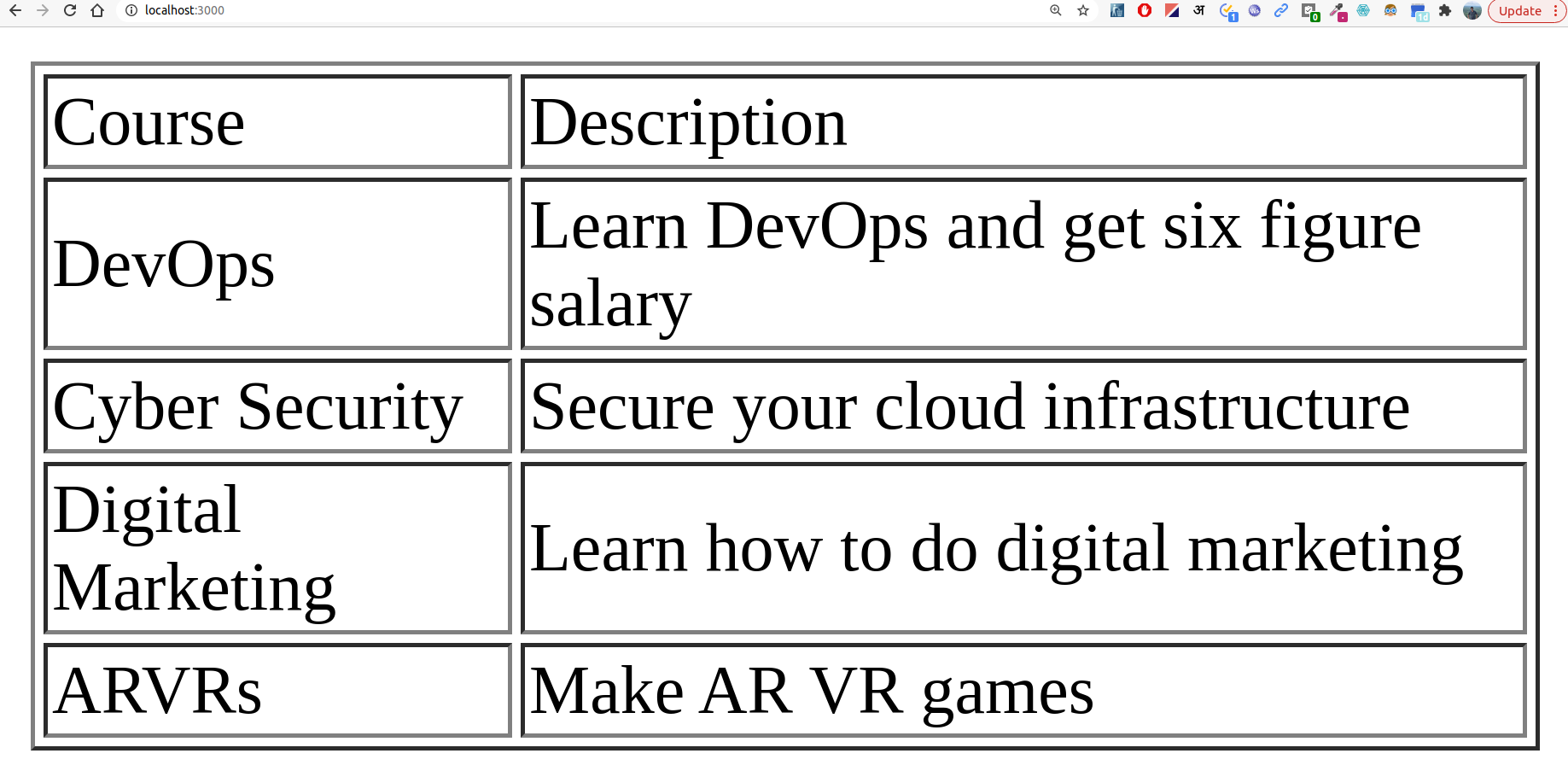
****

**Explanation:** Take the code base and docker build image using CI to build, push to repository

**Full marks** - 200

**Project marks** - 190/200

**MCQ marks-** 10/200

**Time-** One week

**Deadline-** 19th/21st(wait for update) Aug. 11:59 PM AST

**Submission link -** <https://docs.google.com/forms/u/2/d/e/1FAIpQLSf8QutW8O-i3-SLWd01CVXnhrAQpA71b0E_7pVjlhxg2DiNXA/viewform> **Checkpoints:**

1. Day2 **Dojo course** 
   1. Take the code base of dojo-courses app
   2. Write Dockerfile for api with node:10-alpine
   3. Write Dockerfile for web with node:10-alpine
   4. Write Command line for running postgres container postgres:11.2-alpine
   5. Write Docker-compose file with
2. Create two bridge networks for public and private communication
3. Correct volumes for database
4. Api(backend) service with node:10-alpine
   1. Env - CONNECTION\_STRING of db
   2. Don’t Publish Api endpoint, api services will open on 3001
   3. Attach correct network you just created
   4. Correct context
   5. Correct dependence
5. Web service with node:10-alpine
   1. Publish web on 80
   2. Env- API endpoint
   3. Attach correct network you just created
   4. Publish correct port
   5. Correct dependence
6. Db service using *postgres:11.2-alpine*
   1. Volume for postgres config
   2. Volume for database
   3. Publish postgres on 5432
   4. Correct environments
   5. Initialize database with given data
   6. Attach correct network you just created
7. Screenshot of final ui port 80
8. Exec inside postgres db screenshot and command
9. Create CI using github actions or jenkins or any CI tool.
10. Push the image to public docker hub using CI.
11. Push the image to private docker/github container repository using CI

**Submissions**

1. Write Dockerfile for api - 15

*Unnecessary port exposing will cost you: 2 marks*

|  |
| --- |
| FROM node:10-alpine  WORKDIR /app  COPY package-lock.json package.json ./  RUN npm install  COPY . .  CMD ["node", "index.js"] |

1. Write Dockerfile for web - 15

*Unnecessary port exposing will cost you: 2 marks*

|  |
| --- |
| FROM node:10-alpine  WORKDIR /app  COPY package-lock.json package.json ./ ­  RUN npm install  COPY . .  CMD ["node", "index.js"] |

1. Write Command for running postgres container with volume mount database and database configurations - 10

|  |
| --- |
| #docker volume create hello-flask-mysql  #docker volume create hello-flask-mysql-config  #docker network create hello-flask-network  #docker run --rm -d -v hello-flask-mysql:/var/lib/mysql \  -v hello-flask-mysql-config:/etc/mysql -p 3306:3306 \  --network hello-flask-network \  --name coding-dojo-course \  -e MYSQL\_ROOT\_PASSWORD=Helloflask@1 \  mysql |

1. Write Docker-compose file - total 80 marks
   1. Services for api, web and postgres db - 30
   2. volume mount database and database configurations- 5.
   3. Correct network to connect all services - 10
   4. context for codebase mounting- 5
   5. Correct environment -5
   6. Correct name. - 5
   7. Correct alias - 5
   8. Smart Db initialization - 15

|  |
| --- |
|  |

1. Screenshot of final ui - 10 marks

|  |
| --- |
|  |

1. Exec inside postgres db command - 5 marks

|  |
| --- |
|  |

1. Exec inside postgres db and show all records in courses table screenshot. - 5 marks

|  |
| --- |
|  |

1. Create CI using github actions or jenkins or any CI tool and push on public container repository. Paste build file. - 10 marks

|  |
| --- |
|  |

1. Create CI using github actions or jenkins or any CI tool and push on public container repository. Paste one screenshot of build step. - 10 marks

|  |
| --- |
|  |

1. Link of public image pushed. Paste link. 10 marks

|  |
| --- |
|  |

1. Create CI using github actions or jenkins or any CI tool and push on private container repository. Paste build file. 10 marks

|  |
| --- |
|  |

1. Create CI using github actions or jenkins or any CI tool and push on private container repository. Paste screenshot of of one build step. 10 marks

|  |
| --- |
|  |