Docker Swarm :-

Code :-

Maven Spring Code :- {backend]

application.properties :-

```
server.port=9090
```

ProductBackendApplication.java:-

```
package com;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication(scanBasePackages = "com")
public class ProductBackendApplication {

	public static void main(String[] args) {
		SpringApplication.run(ProductBackendApplication.class, args);
		System.out.println("Server Started ....");
	}

}
```

ProductController.java:-

```
package com.controller;

import java.util.ArrayList;
import java.util.List;

import org.springframework.http.MediaType;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.bean.Product;

@RestController
@RequestMapping("product")
@CrossOrigin //It help to enable the resources
public class ProductController {

	@GetMapping(value = "allProduct",produces = 
MediaType.APPLICATION_JSON_VALUE)
	public List<Product> getAllProduct(){
		List<Product> listofProduct = new ArrayList<Product>();
		listofProduct.add(new Product(1, "TV", 20000, 
"data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAALcAAACgCAMAAAB5Yw2OAAACNFBMVE
VHcEz7PADACA6wJwOtLwKLLAD/cwDfBSH6FwT/jgD/gwD+TwCwFgC8AIH+hwD8OQD/dwDFAGbDFQX
ECRCpIAH5PgD/awDEJwPNAFf9RQDsDxFfNQO3BBb/XADBBxD/UgCaIgDPAE38PwD/lQCMHwK0FQMA
Awf/eADLNwL/ZgD0hQDcGgDCGgD/ZAD+WQC5BBV/FAX/fQDVAD0DAwn2IQL6PwC/AHTWCxr+gAD+W
ABoEgXxKgH9UgDkJgD+VQDpHQD/fQDRHQP/iAC7Bhr2JwCkIwB6NgH4NgAEBQrkHQhDfAtlNwLyLQ
DlKgD/eQDCDBP+bgCVeQWeJAAGBQjwNwD7TwDLGgDcJQLaPgH/VwDmMgCpFQDYHwPfagCLSwaGAHF
nLwAAYW7LKQRWMgr/iAD2NwDpQgDaKADLdwL9gQDJCxCvKQD/bgCnFgH/UwBGBQ//jwDzIQDSFAeA
OwHdKQChFADJHwIAuLaxAGL/ZQD/kADnLwD0MwCDTgEfXRm5AInQHAX/kgDkAEa8ATD/fACwAGsCf
FYVORDiBC/XPwC0AEwAAQO3AFgEDg1XKw8AmIK1HgIAhI0yRQasAyt8QgIAk4BkAzrKMQDOIQXOdw
ChFAAWDBMFDw1CAk8BMUwAtq09MgMBjG+2ATSmBhKjUwJvBzSxZQGpZADNaQAwPRNcAmGjHQoAAgZ
JEwQyCgfmGwAHdEeqRwGuADxGBRWeTQCPChgNWCIALDF2HAQTBRQAtalLADIbQ2z0AAAAvHRSTlMA
elk9NoVpR0O0sKKcbm1RoGVMX0JxmEJgnD1Ej19RiIRZgqZBR8iQPo90pZtweIQ5qFCvVUlpW1Jlt
mFXnoCpmnyEe0yUfWlwShxuo29IcVsnjbuLQ6RpRpmBoWE6MIWGhEZMjZJdmDBhZ36GmZGorq1UdY
udiIeEppZ3mFMndnKgR2V6emVKU1d67m9SMVeVhVyRZW6Akx5toX9jm5Z7c0RsZFxxPUlQOpQM2M+
YujVmhrlzbV/B059osxamnRoAACAASURBVHic7ZwJV1TXuq6LnihKUxSFICAGBFSKRhpJAYLSpBQU
CQgiEoMC0qqARAUVBQIoIcSgoqYxdoBwYnBLzjH5c/d9vznXqiKa7Ox79s3YY1y+AkLjIA8P7/xmU
2thsazXeq3Xeq3Xeq3Xeq3Xeq3Xeq3Xev3/XF/+39a3X377p/Wxevwb65Hxzrfg3vNW1bm35356e+
7cube/nfttzzmP+sfa+ugjvLjrnOcHH/3Pn9V//c9//dPagccnnrVnz3+/U8s7wP32O9bLe6znz5/
fM955/vyp1OrT1VV+tLp65Mjw8JEjxcPFg8WDg7byvJiYfKmqqqYvFSUo7AwLi6up+c4qgePLVs2
VlRUXNi3a9euY7uOoU6cCAoK2j1zv/9+/0C1VCnfbPPz8ys7dQovUo+lDqFmZ++Mj3/11VdfoD5z1
3+7ue/d++erueaG8QKfPV53/Oy551HwH2k80AxqsHLywZwVAyK/0l1oAoV9BY8+EawL1Ts2yfgJn
fQfVapWdXVftV+p/xOmfX4kXAL+Z1xQN8ltQn+/fce3N+ZyHyjbVs1+PNVsd0humnby8thSzWwPbl
TepRtKWDTN7BJTuxjsB20+/5acL9Sv+pTJvlP2vehQ30Uflp03/3CxP5+re+X9E3ie6vP7+mMeHtb
rdaODghfJXbdcOXw8PDgIKihOzUvVWHnC3iqY9RR2AXda7hR4ltzHzu2m/U736VICZirAe1n5qQM3
H19s8IN4Z/ddafE0/dLJZy+JcwK3Opt7bBaSb0NGamsq4TswQZi01IdeeK7Kr8qJsY/xj81lldwpPS
```

```
kpERHHj182uCs20nhFRYEBfszgnvHgLoXu6lNmVB6XAbuv782blZWr4z+K77uSk6/xEO5P2E/E93O
8Nw1wy5IF0BbLktXHYpmzWufw7tPVOYvVbmm2nbVYykss2V7lS/hanr/4FvB2f2CPgpsVAXK+Hqdu
STi5CxR427G23W3CPVPaXTrjzgmpDXC4LgP2m5VnOibi++vPFPZnbu6X3x203DtoeX7vtGXOcnp1F
fSxTUuWuQ4fi6/d4ntkztJhtwTa8EPUg7twzpI9Z6kFt79go52Ae3S0q6srKSWJ3IK+cePljRs1e0
VBAcDxpuBYGwvWZ9zYkhSGRYGXgZrYD1aeRVP3V3fvfn2X1Cjq9vR90BKNWN+btiRa0p88tRy1JM7
NWebqomDb2xvcjVHCPTeXZakFd2pebZU/sMGMNii+u/YLdkRShK7Ll4W8go8CgLcJd4Fwt93fPTPj
Hpd+MC4hZytESoj9IPXBytVFHZO7zMjXkhNP7pcvX+JXj3QvWVYtc0+8LU1LPhYfy5yzDop9O8Dtj
LLUg3sSD3LnLc0d9oftqqq9eK1q9985un9/EioioiiiqAiv5L7sIbxCoAuGCoaGhgooPGhGfLup/c
qkiZeVSUge7Nz54Fm0cN8V31+buj1z8vK70wS3gHvpSSy8zlnCwO2si12SnGjuSouvpbZrzuJ/Fjl
pbwf2XpILd1dSUpG7QkJArbnZDgldMbQF1ENtQ+Buo/AZ1bxJLjkBtIAr7AdXF382fH/9Pu6XL7+z
WL6Ltpx+bpk7sGSx9lrmfC2WXMuS86zF3mGZGza5bXNLwp16ktwxVXuD96LADd1nkqg65EZR0Q1iE
9xAlz4+tIV1nOhICxMu2IZt+BZolMLeuaB0f+XGfpd7GkPRcu+qpenJnMXlssxFWSyt4HahnViSy6
O4AGsFd95hi6V29DA/bKfvvaraN4/uP3PmCohDPrxx5QYqIyTjcgaxNfnQ0JaKjcIN8AIJyn1tu1q
PSr+yMg9sgC9AN8DvfnX3h68/98A2uF+yTk+nP/803u5tdUVllURFBUYFBkcFZjldZ32b65zl2SdP
tpaX5+f7+1dV+Y+OttfWtvv7m9h7N2+G7isA/vDDG6oyzmdkXDaFb+ED0/4Wmf6VbnZCE9mTW2P7P
5ifVtz0/fnnn7u5dwi3YH/KGsvNHcnNcmZllWRnZwcHB2cFZzmdjU5na0yeMy+vHB3bf+fO0VG87v
QHdnu7qXvn/v1XwP0hi/Qfnj+fsWHD5cvG2CS20r2F3Crf1L1tDXWfGxu6XwBbuH8gtcn9Vvs2sT9
9kmvNzWKRu6SE5Fngbo1pjYlxyvyYujmV1Ds3+7dvdnPvZUwMbKnz589v2OAJDnLYFvQt0gsx9eiV
oMndx6m9TGNvroJu8f2rcNP3Z589Et+frOG+B92KO9iTO8tprJ3yINlffVOWp2/G5IYH9QcfKOwM3
cQlKTovFYK9+34psfko26Z0HyK4xt7ZvtB0TbiB7cH96BF8m9xK91RupMIWbpAHMzNZ+dQdk4cHsD
fD9qhwQzjyfVi4d67RfZ7cBrm7idM4wzKkdQ9Ub5NC2+4EeB/Xf2/euHVfu3YN3L8auj8HNrY8AP8
d91hkbu6EYAdnZ9cCO5sBb81vzY9Ry1X/dnIbujeT+zDLzX1esD9U2B8o7o1rSmNLTLZt61TUUN4J
7NnZWcHer3Vr7h9Mbm7VHn3/6Ps9ivtTxT01kRvvyS3YWdzQcK0a0x7zDvde4dbxvuGZkjXgbnTRP
aS4BwYGgO23TWT7lRH70MqbB4buF9eE+ys39tefYc/6iMI1t6F7In6ixMCu1R0lS23EqJudz3/zZo
P6fdzK9we63Dm57Mkt2P33Bzo7IdqP1PCtdK88kAln52alW3F/rrnvfgFuARfuT4x0x+eSW7J9WHP
jp9BbSOjG6gm+3eAyLCUmm9kFbyhqU7c74Qa2dEKlu7+fvmG7swyvWvfsCrgfmLoPCreh+4d3uBX2
WCh1lxD7MFJicgeTnEvVmHY2FGL/PicS73di4h6Yxw1y6YJDbS1t3Bc3QPe2zk6/TsHG1mZ29pmRE
uo+qHwbun+4+8Uvv3wrCffgHvt0LD4+PsBVIp0EukFeUlJCbAFXUYnx98xJ+3u5PXSTPMNoKMc3qu
49NARsVANzQupOad2CTd1Mif/eF9cOkhvdW/v+4S42x+9yfzqWMxGZGRCoQsJ4s6H0lhjYGJjYHIh
vj6C4482YKHJP7A+MmScEmx9s3DYq7BpyCzYjLvMkTxyemSlpnz8K7IN6WAo4sSHcIyc7lO7I+Hhw
Bxq6QR5Y4gJ3dnC+Wzi4MWNu3unpm9we3fv8B2vyzZknQm03gQ7fLS0toru4s9jExiZ49s6dZ88ei
G//9r0vwG3Em9w//EDsLxDwxx97cn/6aU58pMEt0LV414XYlLiFs4ProPhrbmnfVZsZE6xKuDA575
kTxATYWNFyw8mNMny3tNTU1DQ0DDDfqDL077LVQ3fu3LmqufFt5xU2uH9V3Eo3g/L4d/mOjA/NDHC
Ru1ZXYCDjHpxtRpyTPcCh299z2mG8uaa6oddTH5jCOS4zMkKQEtkmC/cWcjcMNKCKG4gtKQH3Vc3d
/qBqQXTTN2Ly66863XcV95qcfHo9PhS66TtQ6Y4KDAyEb06amluNS2DDub+sq9SwPNwu8VbY5PbsJ
xmokIiQCCUc1PS9vUGwC4uJ3fdcsE+b3AsL8wY24v2r9G8dEw7Mx2u4cyJDMzMNbFEeJbrV6kq4Wz
U4yAXdbCeMiYDrtclabqZE7e9pvKWnJY7Y2xu8vARbTqXu3BkXbpAvPFtYOCvcBw/KJk10m9y/PH5
8ynNcRoZGiu5A4Ua64dsV6JK1FaOCsdmqcpIqxv3bf8e9Bvu8R0wyNLfKd09c3HaWl5ejGEuSvkPP
Bfu05n6wsGDoFt9cnny+xvdj+D7lya1iElgbWCsRx47H5cqm7xKy52OBxSVhHqnJLbsdFZOdequDl
/PnPfoJsLHpCSkS7h7R3dOjsR2Owc43fauHpO58OT4+fpXgC1eB/ULrxjL25x8lJ8T+5RfxLeAmd2
hoqMJGTjg0s+mbH5aUqJYCcrUIp3C+qF2am9sU/oG5rFIxKXLrXoM9fOTN6qwbm76BPj+/MN+ksa/
JwNTcv4D7F+E+dcrkHpua0rrJSnZ2E/4Uga4SYwPRqs4vAb3Gt463buDn3eNSdH/IfX1RRIScGh5P
QUriCguBXTk8vLKyMjurjooVN8jnr3roVvMluIl95xfWHcn3KTc3YpKp4g1YkGvfgdLCs9SCVvlOT
VW6PbmZExG+FnsDsDNIjUo5nhKRkgJoYtsqK+sEe/bQrNIN7lkKn5+fbwL3ktG/ZcI0scc5LiHc4J
6aioycmNC6A7NFt4CXYHAyKdz1tMpejePS4K7yiIkk/Ly7m2zAy3mOyqKiJOomNXQD2+GwDVbWNTa
uPFsRbMV9VQrYhm6T+9dfgc36RR7APvUPxT02hWlHcWfLCypKCceKJbAki+BOFHOSiolekdN3Lbo3
sCXgCnuDe66UeCcl8cxNDmrj4rpA7fACduvkyrNnXEvNGro194smzX3U5NbYAFc5MX2PjU1hdWLEW
16iWIgJQu/CMoXCnTHleXk2gutxGUPuvf6IyRlJCpj5+GCDO97ELopISqHtLrwUOkZpu9zZOjn5TM
DvfCmjEtx8JbfGPnptmvvir766O37njhoHDItwnyP3MrhD4TtwTcnJTyDHqsvl4jYI+/o85VuT58F
3rYo3uakb1IItOzQID7lyJgk5geqkLmJ3FY6m2hDu8tbJBcV9RwYldJ9GzbtjchTc16Z//pmng+PE
ntXgjx+X6ZwsX7pE3wG/5wY5sAMC4LxXfDvzyvPKDe5UlZPDVTv3799/Rtbf5zM09gcau+gKDzqTU
pJALdijxLaVlzsnDe6rKiWnDXAjJkePNpEbOfkR2Kpfoh7feXyozPA9dmlqDXeUYNcKd0AUuF0uwX
ZzCzb7SS0XVQS/UvRhRobbt8I+c4bYmlqwHV7CDd0LC8C+OntaY7Oio+fnpwV7Drqbpq9dW/zxx8X
x8dnZPpYCP3TIzMmlS1PISUDA+3yLbjRDAS8vL0+1KeE7AR5jciMNN0Iy5IRqg4l9RbDB3bW/KyVJ
Ywu3c3IhGNwciONruKMXlW5iX2tivn9eHL+qsMsOlSnn8P2R4sa0E5kZYPIa2FFRsI0KU/kmdh6xH
XjlASFzAm6eIBddCcnIoOsNOts62rS9f7QL7KAfLcSyBLahe++CcIvuRZM6epHcgJ4729TUND09/f
Pi4vhp6H7T1+fXZ5wfurmnxqaMVVWA7n/KdxTyHUbnpm9wOxypDgRVjUvxnXSlqChEYXuMSCX7zH6
```

```
e54Oe2NuBbbM5W4PJTd3kjib1uOKePnpUbAv3tenFn8fHn6309a32bTOg+VZzow1OTQg24qy5tW8p
NhS3b4dDJSU1ryqbvnnwDbPUbdblG1euFHHGOdMltkk/GkdsiUlrPld9EhPgQrWiBnbTUamzZ0X3I
lOyMrva19dZ5nnQ7Ke4oTtUuDkMqdjkFnIKV+OSvm3kHnWMpqbGVGEfutd/NIncIZ7caCR8vqGIuP
s1dheWJl42LxmVwdnkJjh53eEm98GjSve0gC9GP1t5g5iUkXvbNsXu52f4HoucUGoVatTvsDHv5Br
D0uZIlae0Hakxe2tPsn1zVJI7433YQm1gs8qdzvzghcMLL14IOLmjdbg5Kj11I97RV1dWVt/0bfNT
J6DGUxNmTuInpFWD287BGBjlmRP07wkP3Q7HKBcZ/tRdG5PKHADT4M7I+PDKFS4CIzASiX2G2Ck9G
tvmdGZlZdcDm+DzEg9gm7oVtq9gTwP72crKm87OPh43V2+TY36ehWpuxGTCrRu0wu1rcsdO9K7h5u
IotZ3c2djL81m0ECMnGRtuUHYGsc/sN2QDu9BLdjnlPJjOrk9W3Fc9sKOjp5s090lfX8ZkcXFesN9
0klVd9qHE63xPRU4YOZHO5xmTgERs8yeyVE5sNi+FXTi6uSqfp0NVopuxyFALqRvsiCEhJNYZGe2K
A3ahYNvK+WRGff2LF+jPLxbmE6KjrwI8wXNUngV3k2BTd8cbHrIY2HK5yrZOw7fizgyw4yH0AVE3o
Tvqpvbd25urdXsJ9ijadwy4k7P9wR2hfF8W7CK1owRxkokNatnnANuJtWUzdTc1nX0xH07g0wkJ4j
td6Qb2C1/f9OnF6Wjq7hwmt0I2Llep1r65+AZ1ZsBNw3pAwE1iq48zY3tzwV1XXglur0JJiX8euE/
WBqOdpEQwzgx4xoeCnSHTjcIeHS3kmhvbHOEWbHuyL5YhZ8mt20m0m/ssPu8L7vTFaA7KYWJ3DgC5
u/qiYHdXl6Z5cGdiv6NsC/fNqJvCbSd2bG5uI3wLtpdgs33n156szU+F7pAQDY5oy3lJUVJRkmGbQ
xjC43S6S7IC7D4+vr7CncycRCvb0fycp+70aPTAuuHO4s6BzuqZ0u7u7tKLF0v5KFW+b0ea3B66Ff
ZN/hqoG+CVlZXg3k5sDs4YxKQ239GldYfIFKmwZYujujb/cWGcERPoBjZ0IiVnMTKRk4SrSneCr6E
b3OmoBOiuGy4G9sAMqrv7oq5uN3cudpeK24C/CfCbkpNM4W5srKsTbC/2b1hMjclOTs6OcWDDq7i5
+tPYV7hX4NTeVYgGVChjUnRnlbia7b7pSvfJky98wK2xfch9ltgnk4kdHR220gjs4oGBgWpid1/sN
sEVd3wkz2JJnBkGcBmNQg3uzEytu7FOdPNCHwpM9Sd3cJ7WHVGUtD9JhiSn96IQhR0H7FQvh7btZY
uB7mbqnm4SwBfpCQky4QCb6T6rdPv4wHZC4mQjdA+w7gt2kKJ+uIY7MzOejHpkZmrd+ChzYgLcVui
uq3Rjc7rMTq7PT6VuPIrQq5EUjMgQNhfoPoMpstBh4wSrbdvyYoLhm+EWbhIKN7DDRbdgn8Qn8elw
hd3ff//+/d0z3UFBQQr6IYWb+Y6Pl6TonpIpOWFlBkzEThC7saOucpBXgwl2V5fD1pptr5eYABtu1
fDU2EUa24HViENNOJpbfDf5at8+GpvcvkdVTJJ9wsNhO6y1brChoaH/Pq+w4cV7J048PHFRsA3u+M
hQcscLtujONLkntO7GjsrKQRv7dyF25XFxjvLW+vpsxgSVhBkR1kOKdGfhXiElbnsh+nUes6WwES3
6TvbluPTgTlfY8itQ2OHhiWGTjZWghm5e8LGb2Czl/KJnvpVykZ15M8DUPSG6O4bBjbWzF+XFxZE7
uN4ebJOLwIDNq3xoXpELNpet2NfZvHQ5yO3iuES9OHkS3L7gXqRt4VY/i91uJ3ZvI3TX1PSrS7E09
8MTDx+K8I/2CDd0G0kR2/CeqcAN7A7EpLIcSdHYcQ4nZuvWQmJz8wh4ua5KY2M9IstWTrEG9qi/+H
bZkwHpA+7DJ5PZB4lNbq3b3mxPBPZkXeUa7CCFrcEVN/Kdw6EZr3QT/6YCz1TcHeQuBzYSB26eTtr
AXZJXKNvdFOPquwgdEmDzvJhbBJN71OAOaLaH+/qeZCUnRAt3guI+iZfm5ubE5jBJSU1NTZtc76F9
PyQ6y8x3fCgDTnCBZsW7dY+A+kids264crBhcPv2mpqWGnK7XJO2rpQufc1gksGNDtgj2HFcjTjLZ
Y6FbWD75weXALy5GVOPYCcnSIUn2JXvkyfr7WGsSWel13ZiF7TtPmb6PqHBwa1zsklionuKYHvoZk
o6kHBerNlACy01iECry9UqMyEv0aRu1VkEm8WYOBu1b1mIYUkTLOD1WFcJtz1c2Q4nNwJ/trYerpt
7e1vrbBDUMlTQpi7wNLkfenLn5ORIviPJK9Som5kaO7djhNhHhjF5cYQDPK7Gi0c3Tkchlh4pcV08
QtNXOyalaGxwlztzFXchdpl8biU/OItJITjcQi6YEwxspqS+OTEsDNhOm5fCLjjWplLyPt+bNoWGR
srkE6/iYnRzpXukwwrqI8UHigeL+xvIDd+VrZNZeTYHhyi6opFvSO/pOS7cLdCdK9yFhdhBjG5ub8
+rCqZwKK8X3aAMdw9LfKa2Htixk63OcoVdsKtA+T5B8iAj4ca43JQTCvDIyEzVWDLjI03uXPoG9pE
DB3jdd0M/W2p/TcNgXSt829BbCrvIniIX9OJtjzyr0NPTAt2NWLVjbVAYB+6dvDooJkZxlwh3veIm
to8aqPWJ1A1sG8LNi/UKjhUINsGDug3hJvemnJxQPjFFeDCr+RP84I5sHMkF9pEDxSAfKO6Xp3r7y
d2bBSvbC8V4ii4gR5D7+FALRm5uby7XBoUSE2BXxcTkC3i2XXQLN6CxzvIR3UwJxqSkRLCV7xPMd1
A3Hgrc7RtBCc1hWwE5sEPVMM0U3cDugO60A2lY5PSraqhsnOzlt+fApG5JirpmHW+2bGnhCOjF5FG
JDRL2x/s1d4w8qehK9lXccI2lH5bbopuNO7bXSInEZJe6YDxIgStuTJiGb3DnhBIbDRHYodp4fG5k
rrKdBvC0NHCTfKCheLgRo4fR3W5id+kbG3qObxnaAt11ueDGTCXc9M27IGKygoOxv0xG1YPbLotxg
JO7VlLSSOztwK5Q3OpSd1B366bi4fv6ptCcnEigh2L+UdgCbmILdZqsKXnlSHFxneLeroMS19Xlvi
FjI5/Lbqh09sbmdgyDexS6R+VaWl4sjpbisvskn6yvrceUjmSTXLibaVs65/aWlqGKCmLDt+LuDnL
3QrOfXM8J3RQaSexQhkTeSGNci42X/oH7A8Ae7vAO09ykLuwy7iPp6ZFLelpqBqHbzY3uLdct5+cz
4C6gUrc9OVnpfkFutu7eXuy+JSVykb66DUXlpDvIvbbqdo9L6A4V3fxPpA4KsJ8Qu1hhw/ZWKge3t
Tes11nZINjccBbyPbm3YaNcOrC9Er+QWOtwcYNwo5so6nxOmQgHcwJq5RrKX6AHhlG3G3tfxT5194
zOiTllenBTdGioekvnkcw5uEeIPZwmurduTRuYGdjaT+467zBXL3sFcCHbwaYh3FtU1XhVQrdwN6C
b7Nzs78ZmSuDbrrGbpkmOH6NZp4ThrlC3z9A3G2GQTJjG7HPR3U8k3TkMNv9jjMxIYItvhT2wdevW
3eJ7uKMXv1P2kwZ5Xk+mcgZFUUtMoNu7Q+70Kdy/079dUeezlzAlWJtIRrB6baJuxAbYuSrcCvsCb
585pgIepOqEkHv4lnST28BW/XDkyRPp3EwJoLfOMChpxewmiYmxTt5aUlgozyGAG+tENBK+MCbW2N
4w7yPFAxDuGN3ZXmVwIyTJukQ2sH19fBS2N9umYIP6wgUJSoFanxjkarvmwZ2Ts0kR58jIlKwobOo
+kLZVl4p3Yyy41cbN5jUIaodXQyF77pBgtzTU4RcSaz2ibq1K3cy7IILzOcvXMyPJTEkyZftq7GRw
E9vGCYe6L1D4rn26DwaZ5N1ruHM0N5znqAYeOhJqcKdRODOydYYZB7ZwI988hR8ctA3aHPgJsE5ED
bVITBpjTe5BbHSQbplwgB3lG6W4fbjNbOKoTK6fB3asFVsTLzUmL7D27avYpXISpAdmkGw0L3Z7jM
scWaTojoLXqamRkSnk+4CKiZI9A93FzImV3I2V2HIOVip2R4NwD8G56CZ3h+K2peZV7Q1mIylphmV
SqwnnrC9ffXzs6IkYDfh+g9vd2Bf0TVbGApzUmHy6Wea43LRJc6OP0PbU1FTo1JMp6MZcWZwmYzJo
```

```
q9KNdqK46+p4hFVpG0y1eamFuZBr3eAePoDFb6WNN1UFZ7uwDrQLN+YbnhGeZNf2TRbsMFkTDKoWq
LEvGHfjKewZwRbuGWP9LR2cLZALQtKDeWQKMbml8q1tAxuhQU7AHRabyzMVhGXQJhs4JZzgDZXeYW
GJ9D08PIxtaV5+fnZ2totby2QZlc2y5ZH2l/zC3kzdve6UgPmV4fvYLoObcyaPf0Bu7huAzSUhe4m
pGzXyxBOblSYpGe6wxoYlots2NvKQFuSDGlzIRbc9Mcyb27vhSlt5K2zXc7uAEYhkw7ZemPDjZGK7
uJSpxGpGsLEfflXQP6BuIhwYSDtWLRcadp8AeZCcW3nMl0q3mi+F+gmE35LmvdUAR9CJXdzhzfj29
qpTw0EpL0N4DXUnhgv3cF1dua21VWG7VAsUcG7q0318wu32+maFXVc5bLEsEfsVmF4lWyz9wo0Pjv
mqG4qXnp7gKSE6uMmtdYdKDxTuJ4jJrVtqabJ16zdbvwkiNz4eHj6iuAXcqbg5QpXvGupODA9PxLT
DHNnyoLveVc+TEyLbfex2WVD5+GIVS/dIEL5R5WCxxTLHfIB76ZVduEGOD475GLdCL2FcXvRcx+aI
ceWc2LdukfzWE3J/Q+7ub+j7AMN9oIPcsQRXx52qjD1zQ2VuGLYw4WHezH9eal7+ZDaWrNAthYQrb
NoObw5vdmndg/3CXaF8k/uCB3fi08Q5kqN7nzD7t4xLNkNFD+4xgN+C79daN7G/gW6sVo50dFi9Yw
nOmbkRUitlfErCe2pq0AQTeVIWKylq5aBkgqmbwpMZaoUN7sTm5npXGHVjD+j2bXLvIvcrcHdilq9
eslh8H66ZL9m/dYVOjU3dugTsKR0Tmv4GQaHuYSxrNbckxdtbTmo1OLfMiInoFm5nqxNbs3qEwRXG
wWgPp2mEPNwnAdyJzfbmsIWwXm9r3XBxf9u73BW7DN+du9kH8ZXuh2u4FbrMnGNTY2O3IFxKllSUr
WIi1B3ehm9v3slL5XWNdTooHJXhPunh2Lkg/HWtMfnZzUyx6LaHi2kRDt2J0I2fqNfagcHeUgDuVx
7cba/UuFTcfM7hRLjF8vRhd3fpGu5NOUo8uOGbuuE77bXyjaAwJkc6nlqhG9wyMlEQrqpOccuoTEe
8Jxudja0o6GbjS7SHy86dxYAzJeGJ2Jq5oLtjuLhm6Pfc5mA0fM90ryLofHbK3U8EWnFD96VLyvaB
12kqJ1sFu1NsA7vXO1YXjTd2NFobG1VQZFQmpCeQGzFB1XM/BmztGsA+HJdoJcS2h4XhOyAl6NzMC
WZJow+u5Z7hXY9HLJaE7tLuaQKPnAAABSdJREFUatP3dZN609gYdSvw16/TXivqrWgr0P20w2oFty
YPE264xqeswt3PlSD26HKATerJSeEO4OmwUMsDyn0Em1th6m7ARLmPvvftM/v3nJQH99bSp/BdWl3
q5jaK1sfEtyh//ZoxkaCwdx9RN6ezcpVx71hJOLhzJSgNg9ZY6Pbg5qiE7malO1zCkq67oIDHUncN
sHcJt9u32Qd3CffW0pmL+CmeenJr3WiF2rcKCrC/Udhqyll92iGyeVu9t0dOrNDvrbixgwM2uGMni
d26UN8s2InhmtqHX/al8fCERJlWsdwl9r59bHkaldwNvCnZzY2CbgufLjbyff060a9LTK6P0bcmp2
5pJqJ79QnvqFfYVqvJbYAzKJVKN9sJuGNaJ4ndrLgBnRCezq+S286PwB2rUrIL8w1s5r56xW7te0F
xFxSYfTAoqDQRsLHUXW2en+SIc7697vZ9ibrFd5rGHhkh58iIdcSqgqKw+Zbr58FB0T1N7knxTe7m
5sSARM78iXzCKV1xk5rcWncB9u8XsECxzNkxuSAi4C6Wm8DJjfVJk69Ml03yZzDSTO7rdC7Y1yXel
4RcuGXGAfetA08ADGqrPLy9Mw1uSQ2CMlyJBS4G5XR6AuItMZkU7MQwSbKcvgp3ujr4hm5vbEL72w
oq+CckXhnrkN4L4lvu/9bcqhJKt1XzJkI393WJuAf2rddKN8GV7ltPnwDQOsICubcnNrkxeXSIbsV
N8MmF5vnmMJ5WGtwKO109zZDIdA8X3+efBBDwYUqdK75QsQ+oxbxpvcC9HlxK75QbH6vd3NcVt/zX
9G1wSwtUupETA9sa762aCX4UPVjR3KF7GtxsJwSfxOweporOyZ3gBgc396ADbcDepclf8YUb+X3yR
wL4REmb/E2G0mp9t+a2bb/jFuvXL103qE3fGJSv4VsBq+oY0axqjFqNH0HpRrzF9+SkiS3KE4ycJA
h3YkKYtcPAJrVR/IMMFRUFQ/pPG/DMRq6dUfWO7+ua+7qh+7UH9hPUyIgbe8Q7XjcWq+Eb9ImiW4b
lJLHDmsP0egA7pHCTW+lOSOSg7C84dmyXEm5Q79ql/yLDkNw52O9J7Uft59b4VqV1f+PW/Q24bwn2
E27yFfiIBMT6VLifKvdKN7gT6XtyMixsPux33MwJX6ITEq6yBx7ob9PY+zyxBZyuW3j+2zkgV1gp2
X5rcoLebaC7k6LAX6fRtuBGjkRGxufq8paXXHTEXm/ZRcSG+TRNHz3a1OSLLXo2Nmf1drbBANcEn8
6184mFdHcl+NiR7oG2Y7sL3Nxy1qOzXdDGW9lqeCebugGP2KJb+V7+7X3Fv5z0m/EnlH777dxfqZ9
Qb9/+JHXup3PmO+fMD9YWvvDRP6lzH73vf/SPt+De8W4tLy+/57P/Yn3y7mf+sPZ8sucP631femt5
X43fvn370Xu/8q/UtWvT1679r7/LHrAs/bV/enD538L976k9t5d3vN/uO7Xj9m35g1z/EbWI0O74S
xLxm1m+/e3/a56/XG8J/vFf+He3b+95e/t/H8x/V325/PGO5R3/1OOj27c/sSwv/x1Ef62uLX98EO
Djf/6vviX2wdt/cST8LbVjj2V6efmTxT/7N1/evo0ROX37T//R31yPQLQI8D+J7rhg/yfW4vKOPX/
YxqeXb+/4iz3+b69xgP/Bl64t317+T8VmX/mD+Wdpx463B/9mmH+lvnz7yV9o4+u1Xuu1Xuu1Xuu1
Xuu1Xuu1Xuu1Xuv1n1v/B5SczVvcicYUAAAAAElFTkSuQmCC"));
             listofProduct.add(new Product(1, "Computer", 10000,
"https://encrypted-
tbn1.gstatic.com/shopping?q=tbn:ANd9GcThlAjfwekVFuVXxPlqGVflSMQczaLfpEtxqpIPb
OpJePdXVBksGswRI8HPdeb6T1UtWJSCBRuBcQ&usqp=CAc"));
             listofProduct.add(new Product(1, "Laptop", 100000,
"https://encrypted-tbn3.gstatic.com/shopping?q=tbn:ANd9GcStyUqmRZj9nV--
4CUeggkEaxPF60K_1GxmrZVqyq0H1Smdy15bzLd13-E8sB9Yrb3SlzBeQWQ-Tg&usqp=CAc"));
             listofProduct.add(new Product(1, "Hard Drive", 2000,
"data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAAQABAAD/2wCEAAoHCBIUEhcUEhIXFxcXF
xoXFxcXFxcYFxsaGBcYGBcYFxsbISwkGx0pHhgXJTYlKS4wMzMzGiI5PjkyPSwyNDABCwsLEA4QHR
ISHj0qIiAwMDQyMD0yNjA0MjYwMjIzMDUwMDs1PTIyMDAwPTIwMjIyMjIyMjAwMjAyMjIyMjIyMv/
AABEIAPgAywMBIgACEQEDEQH/xAAbAAEAAgMBAQAAAAAAAAAAAAAAABAUBAwYCB//EAE4QAAIBAgMC
CAgJCQYGAwAAAAECAwARBBIhBTEGEyIyQVFhcUJSVIGRk7HSFBYXIzNyksHhBxVTc4KhstHwJDRio
8LTQ0RjosPxJZSz/8QAGQEBAAMBAQAAAAAAAAAAAAAAAAECAwQF/8QAKREBAAIBAgQFBAMAAAAAAA
AAAAECEQMEFDFBURIhYZGhBRMVUmKB0f/aAAwDAQACEQMRAD8A+zUpSgUpSgUpSgUpSgUpSgUpSgU
pSgUpSgUpSgUpSgUpSgUpSgUpSgVrkkCi5IA6ybDXQVsrkvyj4qaLA4qmqLA8ZhieNWWLiwq5ySzhLB
```

bHNox0tQdPFOr3yMGynK1juNgbHqNiD5xW29fMuC+DxkQefHSyGSRyxiSQ5EuigtIqnKzBQBlFwNA
ATa3T5RfT+usUHTXrFxXLPMobLYliLhQLm3Weodp0rat/FA89B0t6xcVzmvUPT+FNeoen8KDo7ilx
XOWPij0/hS58Uen8KDo8w66XFc75h6fwrIv1D0/hQdDmHXTMOuuf8AMPT+FZ8w/rzUF9nHWPTTOOs
emqHzD+vNQDsFBf3Feq5/L2VkL2UF1NIqKWdgqqCzEmwAAuST0ACspIrc0g1x+3cJ8IheBJZYmZCV
kjLBNVuVcAgOpXQjtNiCDUDgJi9pnFyw7QUARxXjYKLPd1GYONH0UdRF9QDQfQ6UpQKUpQKUpQKgb
UHIHYwt6GH3mp9QtpnkD6w9hoKHEQZzyJXQhhm4sprlvyWzA239FjuoAkEW45UGmpZmJPWdWdmO87
y1VuwNgnCvK5lMnG5QAc3JClmsLsbC7sbCw1NTh85L/wBOI2HU0nhHuQafWZvFFBswUJVSz2MjnM5
G7N4o/wAKiyjsHbXp4M1szNe1tOSD5hW4ddZFBH+Br47/AGjWDgl8eT7ZptLaMOHj4yeQIu4dLMep
FGrHurhtpflDckjDQqo8eUlmP7CkBfSa209C+pyhnfVrTnLuDgF/SS+sNeTs9f0kvrDXy+ThptBj9
OB2CKK370NdDwbl2njIpJRtBIkjbKxeKHoUOzEhAAoBGp7a2vsrUr4rTDOu5racREuxGAX9JL6w1k
YJf0kn2zVCcBj7EnbeHsMhJ4qHQS/Rk6aZvB6+ivOGwmNkfJFt3Du4vdEjgZtN+gF6y+zH7R8tPue
joxhB47/bNZ+CDx3+2a+YbR4S7RhkaP4ZnZGZXKxRABlZlYax67r37axheHWOQ8t45R1PGo/emWtu
A1MZiYY8VTOJy+oDDDx3+1WGwx3rI1+okEHvuK5zYnDnDzEJMOJc6As14yerPplP1hbtrrK5tTStS
cWjDel63jNZa4pfBbQ1vrTJHcX6q9JqP6uO6s10ZtnAsbyy8WcvzWYcWMpJsOTmyk2uuaxsBa1W+z
+ee4+0VFAP4/1uqTgDy/MfuoLOlKUClKUClKUCoO1OYPrfcanVA2rzV+t9xoKTGzMqhY/pHOVOmxt
cuexRr2mw6a3YbDKkYRdyi2upPWSekk6k9JNRsPyppWY6OPVhXF9qlsaRXmSQW5FJUdqRiBQPN
5pDyY1uQN5O5VXtJIA76lE1iiTZkOJQpOgkQEMFJYDNrY6EX3xmM8kTnHk+I7X2rLipTLdqToqj
movQi9ntOtQq+wYzZWxoplgfC8puLuQshReOdo4g7A2XOysB3a2qIsOxTEsowEuVyBGDDDDOGlzRvLm
jDEZ1CIzFt1gOsV6ld/p1jEVnycFtra05mXymvoXATK+zsZAJ0jkkYhS7hbB4lXNvv0NqOqrvaOz9
jxRxSDBcaJlLxLCrM7IqcY0gBYckJY9eoABJtWqPDbGbECAYFuU6RrKVIiLyQidvEvnzBihvqRaqa
28pqV8OJj2X09talstbbDlGUrtVA3F3kzTGzTDjAjAA6IocWXpyLepuydmKskEmKx2HkOGVhFkcZm
ZxlaSV3csxsTZRYCrX4mbM8kT0t71UuNwOxImdHwqgxhi+h0VFzs+4tO5
5jp8vmm3nVsXiGUhlaaQhgbggyMQQekVAr7JiNjbIRgjYWAsHIRsHWAsHIOQPmsc1t6MO+3XUPDQbEkQyL
hRkVipYqwAAUtn51ymltBfUaW1rsrsv6RERiXXLO0tM5y+T13fADhG2dcM1ddlIfFjvUjq5zwoYt2hV
RbqsDsfY00jRR4TRSf9HiV4BmW2ZmSWiYbtFyBAPXrbW40d1Wb5lxjqFhw5IXOLeazXNSN13BKRxjt
SFHKPsL8dd3LLNwkjZQXm2F2rgRNKGNI/Y8VT+GPDW5USFAJJ08BrX/JjyRHO7kgwS1ddL/G5DWNK/UHeu
oFWmNk0JvsbKyqwdfFcC0C4U9L+Y/kVmLMtEHGuWgyBwDVRbD3fbWqGR6o0yu+BSdoAf/3fD88IB+bxWjB60
YSijvvdw2d7p+K8+DEp+jtrAh3jmxFctXVHmpF/+PT3rZ4QZD5s6B1+W9HvJGqbPwmmd2fe5Pu7dB0eI3lhT
M9i/w0F1SlKBSlKBSlKBVdtTwf2vuqxqt2pvTub/TQUOzOdN+ub9yoKsxzaq9lPdpx0iZr+dVIq1G
6g1VNwG5u/7qh1NwG49/3VMIlxncfbEkWLzszHBcMGkmSGzGpFaqRIQYWp2IyQxumDGrHHqHJbKGzj1Ej
JP8FihiQSIZppElkdMiQksKKVSQu0nKUaNY8kVeYzErHttKWpsGxNUFbbqxZYuNJrWbHwGFOJhGkQN
RWULl5W7NY1TDYmDkInmeRolMREMJCtxWJjWJjWVnlQSfNqzuysbsbAtxaBQSdSTazY7DYeR3jEarI8wHH
uozMggdSVhVHUtIrcWXBOUgaAgWL7WkjSGbFKxe0iXrfl6yBWxcH9OpKLWVWnBnSNW4HKRu4jh6g8rYOxNgM
RxJPLklEUczRzcv2mvda80f6saaR0xA3Rrcd66GkSeTGdXZxqwstPQ8EBxPAETyUkvhLWpHUgqZhMEnkmvDj
KUqQoaUoIMw5R760pwd+nGtKuzteVHAwwpPscmbHLGX8Wn4iq/aAXUiDaDNeYSs2vzF+cGmFhMGkRgKNiXiWC
QSpGOI0XVtTGYHsasthxeEdFfXTrrPwAQUhSmYo89ZJfRBzwAMwlkEuS7TWdfSdTDDcp91PXbSAwfwZiTvIBmA
qIZ0eRp4iAleEvqjW7TCIwY7nYKazCbRkW0xv2Vpb9tvQ3oJDFDT0aU9hO8JaBlxALQKuCcTAOA8gBgdKzfycS
JP2pKjaGPuKawLOX0jsmvmWSPd+Exl12cYFIfhTUlBHpIHcmenJZhgYQoKbfaD4lTvRkXGDHEgT6r2m6EmHKDSz
b8NPvESTyavLZYuW0vZXEXyq5OsZ4HbDTXjQzYYmZiFwFXybxIIUd5ysM4a9fOcvYsXhsOqGKgD53LZDh+Ja
ZHEjZlJYnLGS7Fy+Z2y5vCI0IFjXvB7KSNw4a5APgRLqQBmuqg3sLb91MGU2KPKqcnKoW7G7GwAu
x6Tpqa9VmlSMUpSpChpQ0EObnH+uitCCpE/OP8AXRUPE4pIo3rkbKas7nsUXPnrOUqtcbBwbGqwq7+p
n57GNxdx4EI1mkPUMoI89VdqqOoLYD9rcqR+6B0xzyhVCGuAqDJ9iLISMBjCoOBZgCjMCB+GAbs6FhQeuNZ
TUTDRPrI4tJLbMNDxaLzUzN9gTfrZm6l4GQ4wtNBDjqfbvQT6EYvEWuuH1bDmysmzKfkbr9LD24MI6kUAt
ViAXbQWUdOpUd5A3mgv45Vbd6Omo2KxyTRpaa+Z97ZQ6SH/bJuKRDgJ2o8Lg+Gh7tGfyMqfMFfHXsMUonYqQ
JZI+WQDGFJNwNDdgQLgHQg1bKCXhbh1VGKS2ckCyx30d/uUrG7fhiYKyyEsyqMoTe0kUQ3s
OmZD3A9gPOS4wNxSmaUFSEJCRjOWPJZ+VzhZrZ8262lak2gS6/2qfcrAFIiu+9jmck6p19O/QWZF

9g+FuHlR5FjlAR0QgiO95ObaznTrrxPwxwyIXaOWwjaXQR3yrEZT4e/KCO/s1qt2RtSON2DPLKArX
DrGBdXCg6GxIyHW1+Ue4WDcJsPb+7nmFt0e7Pxdv3+ig34rhXBGSGjkNkD6CPcVlYAXffaJvSO23g
cLcPyuRJyZIozpHvm4nIefu+fS/c2/S+t+E8F2BgOmh+j8Ut7L+mvQ4SQ5rCA8/J4G8JmB9Cgeig2
YXhTBI/FrHIDdBciO3zjQqNz/wDXS/1W7LzNh7YjxcZkjV1AKiz5b8pFcc0kbmHoqvw/CSJnRRCQX
kjjvyNDIgcHuFh6K6FUA3ADuFqCNNzqpdv7N+E4Z4SSFd489ulFlRnXzgEVdz86tCb6iUthXWwFhu
A3C3QBVcmNAYkuhFl5F14zOw0QrmzB82mXLbXfvAsFPR1fvHX91eeLXMWyrmI1awzG26539JqBdUr
ArNApSlApSlAqq2jz/wBke01a1V44cvzD2mgp2+cly+BEQT/ikIDLfsRSp+sw6UqxA0qp2E+aHP0y
O7n9qRvusPNVuByaDURf+urUV4eJHZS8alkN0YqrZCRYshPKU20uLGttq2wx5j2dJoPESEm3X7Ous
bXzXjVeNsc6niraZlCgsD0i+YG4tlO/dVfwg2/8EdY44wSU4x5Hz5EUuI1ACgtI5Y80WsASaiLtRc
ZGgMaCVWZTrM6KOLjmDIqZGlLK0RyXFtbnTlShY4hZLuw+GLlAWyrEwbLcBkViRc84nS+l91hhJZC
rm2LNmj0tAG5LZiVKtbKwGU3todOmqXETxpBOUwyvOpZAjcfGskYRXkORyTGBHI5Ck2NhY62FbwY4
W4NpHWWNIVMDSO2eUnKshZSyvdsrxzq6keM46KDo5Y2Oqvj9RplePQsziwDPztN3QLEWrbjlkLX/A
Lal1FuLaJV1RLkKz84a3G4EMR1ni9s8IFjnZMPBhvg6mBlR3mGIxKTRMyvh1DAsOWUCgNyiL2tp0r
7OxAxkUPwXDthSZJs7NIZAvFpGEylzy8zkE6qVYaAg3C8GzyxYriphnynJnUlLEGy9K7rHvNafzSL
f3/E9OomXwtBqQey1QdmwQqryQYfCiRRYhJwwtmQvmYX3ZRqbc0denrD4aMwO6x4cMtyypM0iEZjl
JcWKsRrbpzMpvvqRaYrZjMxdsTOg3kLJkQcgIbDoGmbvNbcFgTGzEzSyXFrSPmA1vcCwsa534NGIj
fCx5pHGEEbM6i0gGbjiSxTkluSLluTrdha02htOSKTD4cKrSzBiziORkVYjGrkIl2F2kQDMwCi5La
AELKfneao676k4jf5v51HG+olMPQ/9VgnS46RWVqLgJsyyD9HLJH6CCP3NUC/TcO6vdeIuaO4eyvd
ApSlApSlAqrx30h7hVpVTjjy27vuoKHgx/c4u1T/E1XZ3CqPgr/c4P1YPpuavG3UHipmE5vnqHXL8
LeE0sLLhcKVSRkaWSVxdY0VSbgWN2sjHcbWFgSdL6dJtOIUvaKxmXRbbwscjLxmFlkKfRyxPxbrmt
mCusiOosBexsbdlQcRg7RJFBhJEjBZypihlu+fnsXkzcZozZrknMNb18wfbzFiGxWMkJ50gnaK2v/
DjuRb6xF+pK1tPi2MfFY2WRZXKITLIjZ1CEo6s1lblruLA30JrsjY262x/TDiY6Q+sYKJ0TIcNLkL
rZUSCLKwIdpCUk1D5hfeeQ199bkiyujjD4skErYyqU03F1MnKHKNtPB3aCvlHFbStriZVJZEUNiXB
Z5FV0RbtqSGt3gjeK1ZNomJZRPMUZC4+fkLFQyqbLm15wPYN9qcH/KEcTPaX16aAvKORikzEnMsiB
E6LDUstwAbKOnvqOkMzTCY4dw62VWZUzBWZQRdZbWtck26AdSLV8jmnx8a5pJsQinQFpZQGPUuu/f
v8U9RqIdr4ryqb1svvVaPp8zysid3Ec4fdMPJYEfApAbXZgkCh2uAxAD3ubk69ANV20kxN1GEw5iy
l82bDwSK99FYAYiMDTNvBNn6Nb/HfzrifKZvWye9Xn854nyib1snvVP4637fCOMr2fX8PgsUCVxMS
TRMCrRR4WKG55MgfMcQwsGJtpfMpItoTYwwpMEjkwkqrGMyPK4zq4AACusjPnszDNfUXFzXw/wDOW
I8ol9ZJ71b8Ft/GRMHjxUoI6C7Op+srkqfOKT9Ot0lPF17PvWI3+ao431UcE9ujGYYSZQsiHi5FF8
uYcoMt9cpBvbo1HRVuN9cFqzS01nnDqraLREw9CqzYx5eKHViW/fHGasxVXsk/P4wdU6H7USVRZ1E
HMX6o9lbK1Qcxe4VtoFKUoFKUoFU+0Tyn7B/pq4qi2qfpT/hb90dBU8Fh/YsP+qT2VdNuqm4M/wBz
w/6pP4RVy24UHivn3DzCFMWuIckRSwEx1L0zZqKkZ6GpAmp3ikfhMbdEy/7CqC4PAytM2w36qG4yYPBnV/QzFNfpqlE4T4mf+wr8Cy+XRzSV
qNbKSDGdVBUXGpr6r6W+ICx7b+jwAAWwvfcbWmfSSV
gTpdbxZmDZ5X5JIuGBkk5Qy/SCxAryswjRuP5C8kKsQvYchSxkvA0wB6EXYoSnnyfbYLPgjK85f
PxH2b5OfWy+/TitL19oPs842gySA8ZjeMyq7IOKsb6lFabuzFza2+tEmEwtzlmNnFh+bNTzw+PzCp
mBuC2aSx6CM5IB7aifqNOkSRs7dZVn5OdmPDhC0ilWnJF6N1DWMUFlm815dWrTwQj6a72V3Wvs2fnzj
e82nq76VitYrHR6FVGyj/bMcvU8DfFai/CriqPZTf/I48f4MIf+yUfdVnxmYL3VtrThuuYP66a3UC
lKUClKUCqTGpmcwTvPbeEjkrM4yyQerg4Y6bpjmxOrQWEJjnbX4Bc3d1z+UZ+IjuR5y75awisXv
3OW9r23Mp8FhpY9gv0EAbAEHMpNla1jfxHHgt7aDcy2rzWVHsUbHVtAxlebauGaEcsmZbs4p1clO
zSsV5ZtQt7FtwAuxtvvsBrp19FBlm6KzyMpMnpAKxXYcQ3IvoQjtp0g6dNWHNhwqi8jr2TbSy
h/8ASm/lQXwrmODmLWXaWTLUytVSGGizdBZFkzWPnNX0UyTIbJOi6XzxmCnDC+VS1vvUdxG7Q1r2Zsy
OAOI0C8bIZHy7sxaUKoHcTYXoL3CHUgMuD3VRE8pv4rUm9ZtaegtzqgiSMsnmEa4ccQYD8wq2x1ON
rWuJLkObHKuQPYYi3KZegDRA4OeeQ2vTGoRLuZ3a8rdWqpraqquX0dSFQqzrVeWeKeQU46Q+mP7cYn
Lc3A6emvWFlLA5rZlsrFeaXyyqWydYNWvUN8USIBgQnqC09hrKcUhS9aVqZLLtRkB8Uunk3KqvIndk
/wAMUd6SA/w101qxxa+9l+WR/Zk72I8Ff8w6ob8RuosyoSkUAoY2qWmdEfn3SKnJx7+kMqGr1SMnz7
j+zJ7tZ+PWy/LY/ldOkyEGAzZ2MYCBgQPZ8NeRxnRAkTrx8IQe7kaQGeTfnaejYGPBytvLYHhHW/BjJz
7VxBiM6K8eHkZGAZW4Ihc9dbSWrZCvxCLeAWLAkGWSUYpldBusVddaqGP6FmCHNsFx3hLCEZE0nF5gmGbp
PdoOHWyj/AM9H51k9yuiCDqHoFZA7KCBs/beExKBumxNcKJiJNo2y9RkmjRiOvZLiBYZsoBkwHgDCwDF9K+z
uUDeka9A3sd+l82/MaxQer1ilKATXmOVVF0ZBh6ay63GhtfzI7z
ggZgoHJUKvKLW1toKC8wXN85+6pNRcSSoJnDzj5IYHTizaRQ8asQXuW
AO+17nW+/q0r7XXwT8tEartJSw0eCNhYgePIpv180MZMpY5ZGhdTvVoImX0MDWflH2oo+1T/zKjuh
h+9DXn5RdreV/wCCTh/8ABqo/M8vkuI9XJXJ7tPzN5LiPVSPYgrD/wArD/7dYP9Tqt8WTwOfQ7+UH+
3VUdjzeST+ql92rLYPA/+1F2jw12rLYPA/FYyR4oosjgdhMWj5LMVBUFSTqD0dFB7+UPnS/tp9fU/Ul
ovqn9HiNmin/AJd

```
xJeN7Zl4uNb2IYaogO8A7+iov5mlucuGmYBmXMiOykoxVrMFsdQaJsWckD4LPckAXjkA101OXSgre
Pk8d/tN/OpGzdrT4eUTQSMkighW51gwsws1wQR1irHhJwfkwEwgxOTOUEnIcsuViQNSo1upqnYxjo
v+1+FB0fyh7W8sb1UPuU+UPa3ljerh9yubzR+L/3/hWxFjI6Oeq84bje7ancLDs13jpDoPlC2t5Y3
q4vcp8oe1vLG9XD7lUaxxXAuNZMl8w0XTlnlbte7Q8qrrgNsFcfjo4Cp4u5eWza8Wmp3bsxKrp41B
7+UPa3ljeqh9ysfKHtbyxvVQ+5X2X5Kdj+Tv66X3qz8lWx/Jm9dN79B8a+UXa/lh9VD7lZH5Rdr+W
H1UH+3X2T5Ktj+TN66b36D8lexvJW9dN79BM/JvjZZ9mxTTvxkkhdneyi9pGUaKABZVUaDorq6j4P
CxxRrHEoVEUKqroABuAqRQKUpQKUpQKi4jAxSEGSKNyNAXRWIHULipVKCAdj4XyaH1SfyqRBh0QWR
FUXvZVCi/XYdNb6UClKUCvNumvVKDyABXqlKCLiMDDIbyRRubWu6Kxt1aitX5nwnk0Pqk/lU+lBB/
NGF8mi9Wn8q9fmvD/oIvVp/KplKCJ+bYP0Mf2F/lWyLCxobqiqTpdVA9lb6UClKUClKUClKUClKUC
lKUClKUClKUClKUClKUClKUClKUClKUClKUClKUClKUClKUH/2Q=="));
          return listofProduct;
     }
}
```

Product.java :-

```
package com.bean;

public class Product {
private int pid;
private String pname;
private float price;
private String url;
public int getPid() {
     return pid;
}
public void setPid(int pid) {
     this.pid = pid;
}
public String getPname() {
     return pname;
}
public void setPname(String pname) {
     this.pname = pname;
}
public float getPrice() {
     return price;
}
public void setPrice(float price) {
     this.price = price;
}
public String getUrl() {
     return url;
}
public void setUrl(String url) {
     this.url = url;
}
@Override
public String toString() {
	return "Product [pid=" + pid + ", pname=" + pname + ",
price=" + price + ", url=" + url + "]";
}
public Product(int pid, String pname, float price, String url) {
	super();
	this.pid = pid;
	this.pname = pname;
	this.price = price;
	this.url = url;
}




}
```

Mvn package

Create a Spring Maven Docker Image command :-

And add a Dockerfile code :-

```
sudo docker build -t product-backend-app . -f Dockerfile                    //ubuntu command

docker build -t product-backend-app . -f Dockerfile
```

Angular Code :-   {Frontend]

ng create a

app.component.html :-

```
<app-product></app-product>
```

product.component.ts   :-

```
import { Component, OnInit } from '@angular/core';
import { Product } from '../product';
import { ProductService } from '../product.service';

@Component({
  selector: 'app-product',
  templateUrl: './product.component.html',
  styleUrls: ['./product.component.css']
})
export class ProductComponent implements OnInit {

  products:Array<Product>=[];

  constructor(public ps:ProductService) { }

  ngOnInit(): void {
    this.loadAllProduct();
  }

  loadAllProduct(){
    this.ps.loadAllProduct().subscribe({
      next:(result:any)=>this.products = result,
      error:(error:any)=>console.log(error),
      complete:()=>console.log("completed"),
    })
  }
}
```

product.component.html :-

```
import { Component, OnInit } from '@angular/core';
import { Product } from '../product';
import { ProductService } from '../product.service';

@Component({
  selector: 'app-product',
  templateUrl: './product.component.html',
  styleUrls: ['./product.component.css']
})
export class ProductComponent implements OnInit {

  products:Array<Product>=[];

  constructor(public ps:ProductService) { }

  ngOnInit(): void {
    this.loadAllProduct();
  }

  loadAllProduct(){
    this.ps.loadAllProduct().subscribe({
      next:(result:any)=>this.products = result,
      error:(error:any)=>console.log(error),
      complete:()=>console.log("completed"),

    })

  }
```

Product.ts   :-

```
export class Product {
    constructor(public pid:number,
        public pname:string,
        public price:number,
        public url:string
        ){}
}
```

Product.service.ts   :-

```
import { HttpClient, HttpClientModule } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { Observable } from 'rxjs';
import { Product } from './product';

@Injectable({
  providedIn: 'root'
})
export class ProductService {

  constructor(public http:HttpClient) { }

  loadAllProduct():Observable<Product[]>{
    return
this.http.get<Product[]>("http://localhost:9090/product/allProduct");
  }

}
```

App.module.ts   :-

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { ProductComponent } from './product/product.component';
import { HttpClientModule } from '@angular/common/http'

@NgModule({
  declarations: [
    AppComponent,
    ProductComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,HttpClientModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

}

ng build

Command to create a image :-

Dockerfile code :-

```
FROM nginx

COPY dist/product-frontend/ /usr/share/nginx/html
```

```
docker build -t product-frontend-app . -f Dockerfile
```

Create a Docker compose to create a Docker Swarm ,and docker interconnected :-

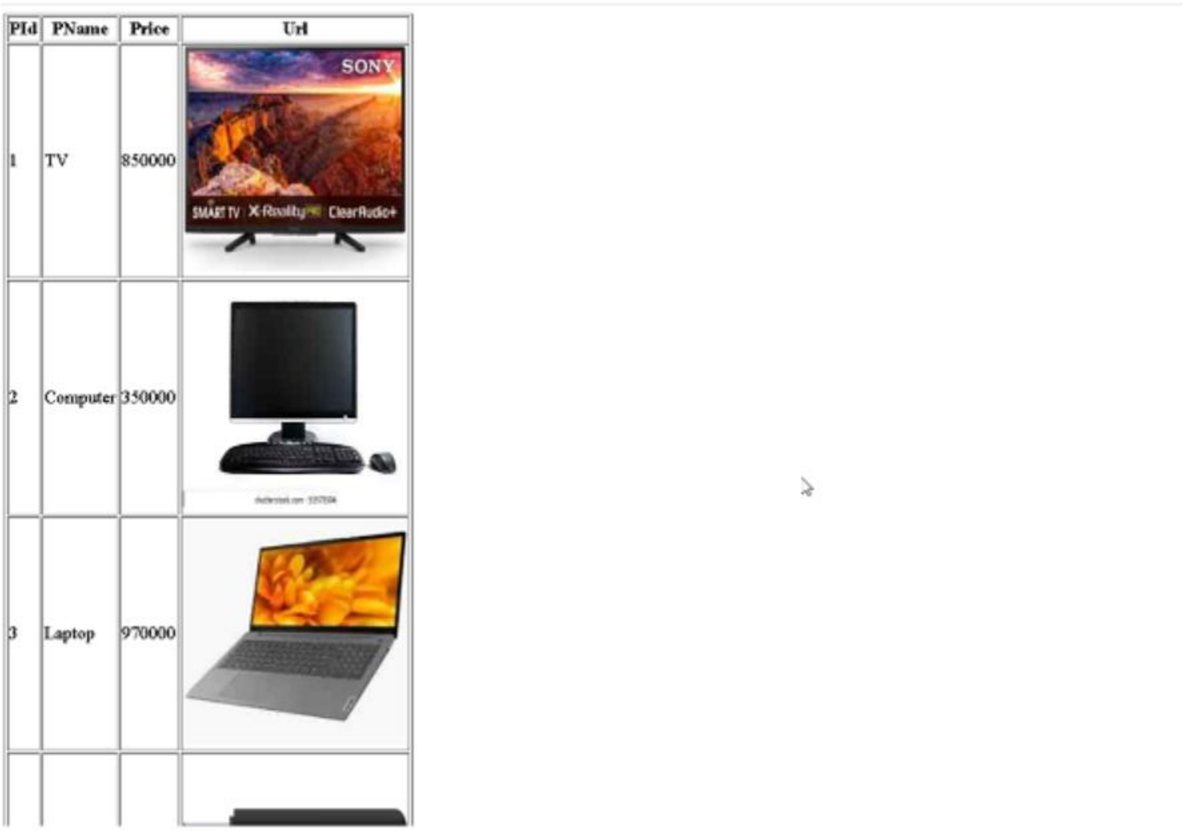
docker-compose.yml Code :-

```
version: "3"
services:
  product-backend:
    image: product-backend-app
    ports:
      - "9191:9191"
    networks:
      - product-management-system
  product-frontend:
    image: product-frontend-app
    ports:
      - "80:80"
    depends_on:
      - product-backend
    networks:
      - product-management-system
networks:
  product-management-system
```

Start Docker -compose :-

```
docker-compose up
```

SS:-



PId	PName	Price	Url
1	TV	850000	
2	Computer	350000	
3	Laptop	970000	