Source Code Entity (Go/GORM)

Backend/entity: Seller.go

```go
package entity

import "gorm.io/gorm"

type Seller struct {
    gorm.Model
    SellerName  string
    PenName     string
    Email       string
    Phone       string
    BankName    string
    BankAccount string
    Username    string `gorm:"uniqueIndex"`
    Password    string `gorm:"uniqueIndex"`

    Products []Product  `gorm:"foreignKey:SellerID"`//
Define a slice of products associated with the seller
}
```

Source Code Controller (Go,Gin)

Backend/controller: sellercontroller.go

```
package controller

import (
    "net/http"
    "github.com/gin-gonic/gin"
    "github.com/NPimtrll/doublesheet-project/entity"
)

func GetSellerByID(c *gin.Context) {
    sellerID := c.Param("id")
    var seller entity.Seller
    result := entity.DB().Where("id = ?",
sellerID).First(&seller)
    if result.Error != nil {
        c.JSON(http.StatusNotFound, gin.H{"error":
"Seller not found"})
        return
    }
    c.JSON(http.StatusOK, seller)
}

/*func CreateSeller() {
    seller := entity.Seller{
        SellerName:  "John Doe",
        PenName:     "JD",
        Email:       "johndoe@example.com",
        Phone:       "123-456-7890",
        BankName:    "Sample Bank",
        BankAccount: "12345",
        Username:    "johndoe",
        Password:    "password123",
```

```
    }

    if err := entity.DB().Create(&seller).Error; err
!= nil {
        fmt.Println("Failed to create seller:", err)
    } else {
        fmt.Println("Seller created successfully")
    }
}*/
func CreateSeller(c *gin.Context) {
    var seller entity.Seller
    if err := c.ShouldBindJSON(&seller); err != nil {
        c.JSON(http.StatusBadRequest, gin.H{"error":
err.Error()})
        return
    }

    if err := entity.DB().Create(&seller).Error; err
!= nil {
        c.JSON(http.StatusInternalServerError,
gin.H{"error": err.Error()})
        return
    }

    c.JSON(http.StatusOK, seller)
}


func UpdateSeller(c *gin.Context) {
    sellerID := c.Param("id")
    var seller entity.Seller
    if err := c.ShouldBindJSON(&seller); err != nil {
        c.JSON(http.StatusBadRequest, gin.H{"error":
err.Error()})
```

```
        return
    }
    result := entity.DB().Where("id = ?",
sellerID).Updates(&seller)
    if result.Error != nil {
        c.JSON(http.StatusInternalServerError,
gin.H{"error": "Failed to update seller"})
        return
    }
    c.JSON(http.StatusOK, seller)
}

func DeleteSeller(c *gin.Context) {
    sellerID := c.Param("id")
    var seller entity.Seller
    result := entity.DB().Where("id = ?",
sellerID).Delete(&seller)
    if result.Error != nil {
        c.JSON(http.StatusInternalServerError,
gin.H{"error": "Failed to delete seller"})
        return
    }
    c.JSON(http.StatusNoContent, nil)
}
```

UI (React)

src / components / seller / seller.tsx

```
import React, { useState, useEffect } from 'react';
import './NewPage.css';
import { Link } from 'react-router-dom';
import { GetSellerById, UpdateSeller } from
'../../services/https/index';
import { SellerInterface } from
'../../interfaces/ISeller';
function NewPage() {
  const [seller, setSeller] =
useState<SellerInterface>({
    ID: 0,
    SellerName: '',
    PenName: '',
    Email: '',
    Phone: '',
    BankName: '',
    BankAccount: '',
    Username: '',
    Password: '',
  });

  const [isEditMode, setIsEditMode] = useState(false);

  useEffect(() => {
    const fetchSellerData = async (sellerId: number)
=> {
      try {
        const response = await
fetch(`http://localhost:8087/seller/${sellerId}`);
        if (response.ok) {
          const sellerData = await response.json();
```

```
          setSeller(sellerData);
        } else {
          console.error('ไม่สามารถดึงข้อมูลผู้ขาย');
        }
      } catch (error) {
        console.error('เกิดข้อผิดพลาดในการดึงข้อมูลผู้ขาย:', error);
      }
    };

    const sellerId = 1; // You can change the ID as
needed
    fetchSellerData(sellerId);
  }, []);

  const handleEditModeToggle = () => {
    setIsEditMode(!isEditMode);
  };

  const handleUpdateSeller = async () => {
  try {
    if (seller) {
      // ทำการส่ง HTTP PUT request ไปยังเซิร์ฟเวอร์เพื่ออัพเดตข้อมูลผู้ขาย
      const response = await
fetch(`http://localhost:8087/seller/${seller.ID}`, {
        method: 'PUT', // ใช้เมธอด PUT เพื่ออัพเดตข้อมูล
        headers: {
          'Content-Type': 'application/json',
        },
        body: JSON.stringify(seller), // แปลง state
seller เป็น JSON เพื่อส่งไปยังเซิร์ฟเวอร์
      });

      if (response.ok) {
```

```
        const updatedSellerData = await
response.json();
        console.log('ข้อมูลผู้ขายถูกอัปเดต:', updatedSellerData);
        setIsEditMode(false);
      } else {
        console.error('เกิดข้อผิดพลาดในการอัปเดตข้อมูลผู้ขาย');
      }
    }
  } catch (error) {
    console.error('เกิดข้อผิดพลาดในการอัปเดตข้อมูลผู้ขาย:', error);
  }
}


  return (
    <div className="NewPage">
      <header className="App-header2">
        <div className="container2">
          <img
            src="https://s3-alpha-
sig.figma.com/img/06e9/70f0/e79d0f7fa6cfd285b65855eedb
d6549a?Expires=1696809600&Signature=FxCCpQ-BmThV8M-
vCB4Prvh2kAvOTuJn5s8huAVq5m3ZOAU94-
iNtCWY9UbY9OAX6F4ghIyWz1neqEjU6IKXLBfjobfpzrwP7T2eVGwt
6NAJyLXF-
IuxsC~FDXHP5cjxT2n23Yci~snXxJSEAHVXdKnNn4DBPXQovNjD5KW
kqHiJFkhV4KqrD~4yi3Ao02-6pHqfASoQ~EwIixU4T-
RswAqj3gk550q4Qdf2OY2HldOxzz7odTAY1Sn0PCsHnAGx2PHuYl5d
U0Wf0Q88v-CDF-
T7f56cEUTmfGvVWd1tVGp1nl7dKHodoFiiF9G~0uW7TRByeSCwHJva
q8dd6UTu3Q__&Key-Pair-Id=APKAQ4GOSFWCVNEHN3O4"
            alt="ไอคอน"
            className="iconInHeader2"
          />
```

```
          <a className="head2">Name</a>
          <div className="sidebar2">
            {/* แถบยาวด้านซ้ายของคุณจะอยู่ที่นี่ */}
            <img
              src="https://s3-alpha-
sig.figma.com/img/3fca/1271/37bd5360d1ce0827d11800eca3
cb9e31?Expires=1696809600&Signature=dyXDzRx6~e9I2beWta
iElFMwu6qTSJlHvMZkKngrTIogcS7MHqEACIemH6BFbs1ovORoWD1X
zSTHWQL0ZLPbdD~-
xqzgeb~c43NEougT~KHnnkbs6pf0Q4Dz46P67IJGK06tKGMSfTqWE~
Ud74plL52438NA-
mXHKunfjO3GiGsK3aC7nIaAUfmggu5pHffaTEBsRcxoeUqrgx23S6i
n-Mt~Jf30XDKGC2gTxI3oNQqucTkLGmHfI-
kN~QuKfpjbiotlY7TsNwWpk4r4AGVNglP5~l-
oKSGHZ1zJIh5zwdQMY9zNU6ifLCR2d~zlWX0oIsDcyE-
RRJsLfPVGa7zytw__&Key-Pair-Id=APKAQ4GOSFWCVNEHN3O4"
              alt="book"
              width="100px"
              height="100px"
            />
            <a className="logo2">DoubleSheet</a>
            <Link className="SidebarButton2"
to="/Home">
              <img
                src="https://s3-alpha-
sig.figma.com/img/572b/d84a/693d97155dcaaf3c78d15fac98
1c7b80?Expires=1696809600&Signature=ldEuxAdxtM~BMoV5G2
LH21z6mUGct~DE3K2BKqe94JViM8WsEjAdMTRw6Mh-LOBImBZPQ-
FShEAna9QfSsFyzjpyiwHNMtSrvf-SM6hq5g3Psl41ojDgVfE-
CxdrUPftaWd9AJurmit5buGB-
42d~qU1oqKl1iN3EDy1jAn0dF24hmRMV05bLoRS5RKCFmCG479WO~W
Ypo5opUIv0raDFsazhsIJG0uq3kyJhcdkt8KQ9EP7X2qwbuVWA3Zwa
xAkndOb9U~CftQBfAWUbMcgoPEiMNvS7lk3eg~DRkBpbGKp~S99Ixj
```

```
oxapqTPfBKKDNIUu0Tvv8b-BiqoVFCcYFSg__&Key-Pair-
Id=APKAQ4GOSFWCVNEHN3O4"
                alt="ไอคอน 1"
                className="SidebarIcon2"
              />
              My Sheets
            </Link>
            <Link className="SidebarButton2"
to="/SellSheet">
              <img
                src="https://s3-alpha-
sig.figma.com/img/e108/f413/fb9aa675c0c5a2ec98034c1d34
09e951?Expires=1696809600&Signature=ErYLAQRD7toWfEcHMc
skDs~2Hwyfv-9W1D8-04KdtfTxLtUO3JEbFmIEVvl-
jwDo10GPsxfkJHgmMIJ67Cvkvrs4vMGmpCXBGVu9Thn0B5yi2yNHtL
3Al2yt~e4TzeioSIWzJBh~5fBC-jl8N8uzjJM-
dkSf4siDlUjw4g8sM03Ln04iCg0K3UXn5WH5pZmTCKjvmAGdIyhoaw
wHeObKIg1kJgAC5T9Z2UtNdUl95W20gzA-
hTlbHBibb99Y6Q6VSsTE~O-
djd5UsOkxRnt6udvYJt5AKl44EtyymoGAgrR9FEETVr~s2e17O1HC~
cHDWH1HkQz~ftnLwROqbngkkQ__&Key-Pair-
Id=APKAQ4GOSFWCVNEHN3O4"
                alt="ไอคอน 2"
                className="SidebarIcon2"
              />
              Sell Sheets
            </Link>
            <Link className="SidebarButton2"
to="/History">
              <img
                src="https://s3-alpha-
sig.figma.com/img/8e6f/2d0d/7b119cadc181ef7a1f7505aff8
0dd942?Expires=1696809600&Signature=o3WlYjSJwliKypKXrm
VZ9XeAj5wS0UCBsO4BGdScfwlmyEhfDSTYI7omRU5tZ89-
```

```
Hfqrew5TpM2CO8ZaZHrAb1kjBK1q66-J5UokgyBx9L4Xpvyay-
XzZqO9~EjrODfxX61WzfIuw6GJperYmPm7s6BQOE-
UUcX3yhQAGAKiYuqvtSZnIIUefg8FNPzZHgyJl4K6XlnlwU~BDoQP4
Z8YcALon8QXbeCZWT6dOCsL8zomSVlmpz-
vWxRCI6OJNUZ24I2bYbZ90EuUF1G70SIWMXPm2MyaWluApNLYG4DEU
YdJ6-Gw4a1WTmm14dQUQA5Sf8omTf76cy8Nj31McoDivg__&Key-
Pair-Id=APKAQ4GOSFWCVNEHN3O4"
                alt="ไอคอน 3"
                className="SidebarIcon2"
              />
              History
            </Link>
            <Link className="SidebarButton2"
to="/HomeCustomer">
              <img
                src="https://s3-alpha-
sig.figma.com/img/3ef1/0cff/0b423fa0e5eed088d433a6587a
826b9c?Expires=1696809600&Signature=RjipMKuylWYzAGa8FR
PTMqbBEeyjDys8dc4rfCyHXhzBQQ2izuMHLJXd77yNhMxj9c7CII85
avDJKkkW2uUJWsu9b7iJvkuD8-
BJIyGUbEnzskVEJh5FkWosa1KSvp1~IIrnNJfBBZVKQMlpl3~ex0hV
YmvIyDXuUhSq~2hlLNUIdwzRbolSBnymveVGPb7jyg158tvpnsXRqr
F-
3iYlUBIUiV0dEvE0whn~sT0u2JLUR1PTkzhyeImuL9gOavqasUTfnG
XyFxaulPv5baoGvOOGKgtDJHgFWLm1-
WU0zqn2TjXmbXu1o7V216tETd4LPoq-
tChsMpjHo2mDZotelQ__&Key-Pair-Id=APKAQ4GOSFWCVNEHN3O4"
                alt="ไอคอน 4"
                className="SidebarIcon2"
              />
              Customer
            </Link>
            <a className="SidebarButtonout2">Sign
Out</a>
```

```
          </div>
          <div className="contentbig2">
            <div className="title2">ข้อมูลผู้ขายชีท</div>
            <div className="content2">
              {/* เนื้อหาหน้าเว็บไซต์ของคุณจะอยู่ที่นี่ */}
              <div className="inputContainer2">
                <div className="inputLabel2">ชื่อจริง</div>
                {isEditMode ? (
                  <input
                  type="text"
                  className="inputField2"
                  value={seller?.SellerName || ''}
                  onChange={(e) =>
                    setSeller({ ...seller, SellerName:
e.target.value })
                  }
                  />
                ) : (
                  //seller?.sellerName
                  <div
classNameName="inputField2">{seller?.SellerName}</div>

                )}
              </div>

              <div className="inputContainer2">
                <div className="inputLabel2">นามปากกา
{seller?.PenName} </div>
                {isEditMode ? (
                <input
                  type="text"
                  className="inputField2"
                  value={seller?.PenName || ''}
                  onChange={(e) =>
```

```
                    setSeller({ ...seller, PenName:
e.target.value })
                  }
    />
                  ) : (
                    <div
className="inputField2">{seller?.PenName}</div>
                  )}
                </div>


              <div className="inputContainer2">
                <div className="inputLabel2">เบอร์โทรติดต่อ
</div>
                {isEditMode ? (
                  <input
                    type="text"
                    className="inputField2"
                    value={seller?.Phone || ''}
                    onChange={(e) =>
                      setSeller({ ...seller, Phone:
e.target.value })
                    }
                  />
                ) : (
                  <div
className="inputField2">{seller?.Phone}</div>
                )}
              </div>

              <div className="inputContainer2">
                <div className="inputLabel2">ธนาคารที่ใช้รับ
รายได้</div>
                {isEditMode ? (
```

```
                  <input
                    type="text"
                    className="inputField2"
                    value={seller?.BankName || ''}
                    onChange={(e) =>
                      setSeller({ ...seller, BankName:
e.target.value })
                    }
                  />
                ) : (
                  <div
classNameName="inputField2">{seller?.BankName}</div>
                )}
              </div>

              <div className="inputContainer2">
                <div className="inputLabel2">
                  เลขบัญชีธนาคาร (ตัวเลขเท่านั้น)
                </div>
                {isEditMode ? (
                  <input
                    type="text"
                    className="inputField2"
                    value={seller?.BankAccount || ''}
                    onChange={(e) =>
                      setSeller({ ...seller,
BankAccount: e.target.value })
                    }
                  />
                ) : (
                  <div
className="inputField2">{seller?.BankAccount}</div>
                )}
              </div>
```

```
            </div>
            <div className="buttonContainer2">
              {isEditMode ? (
                <>
                  <button className="actionButton2"
onClick={handleUpdateSeller}>
                    ตกลง
                  </button>
                  <button className="actionButton2"
onClick={handleEditModeToggle}>
                    ยกเลิก
                  </button>
                </>
              ) : (
                <button className="actionButton2"
onClick={handleEditModeToggle}>
                  แก้ไข
                </button>
              )}
            </div>
          </div>
        </div>
      </header>
    </div>
  );
}

export default NewPage;
```