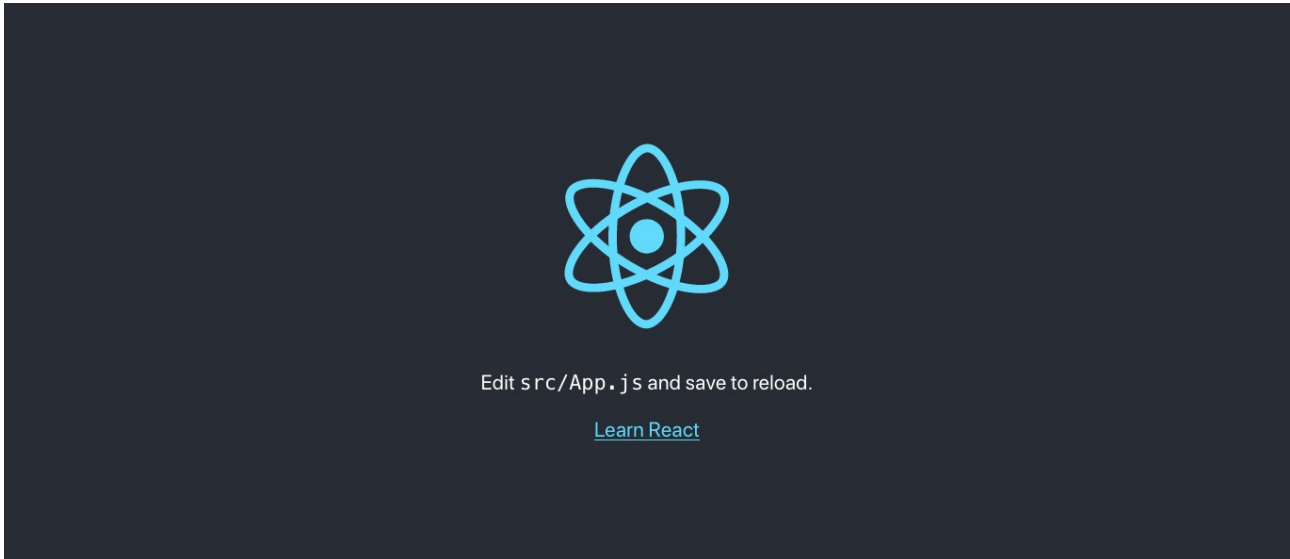


React Project Create

```
npx create-react-app first_app
```

```
cd first_app  
npm start
```



React Project Structure

0_install

- first_app -
 - node_modules
 - package.json
 - package-lock.json
 - public
 - **src** -
 - App.css
 - App.js
 - index.css
 - index.js
 - reportWebVitals.js
 - components -
 - header.js
 - footer.js

1_hello_react_app

- src

2_components

- src

3_props

- src

4_events

- src

5_state

- src

(5) React State Assignment



Copy 0_install/first_app/src folder to 5_state folder.

Create components/about.js file.

```
import React, {Component} from 'react';

class AboutUs extends Component {
  state = {
    name: '...'
  }
  setName = evt => {
    this.state.name = evt.target.value;
  }
  clickMe = evt => {
    this.setState({name: this.state.name});
  }

  render() {
    return (
      <div>
        <div>
          <p> {this.state.name} </p>
        </div>
        <div>
          name : <input onChange={this.setName} type="text" />
        </div>
        <button onClick={this.clickMe}> Click </button>
      </div>
    )
  }
}

export default AboutUs;
```

Change App.js file.

```
import './App.css';
import AboutUs from './components/about.js';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <AboutUs />
      </header>
    </div>
  );
}

export default App;
```

```
cd 0_install/first_app
```

Run react server.

```
npm start
```

Create 5_state/assignment folder.

Copy 0_install/first_app/src folder to 5_state/assignment folder.

(6) React Life Cycle

Change about.js file.

```
class AboutUs extends Component {
  state = {
  }

  constructor(props) {
    super(props);
    console.log('constructor call');
  }

  componentDidMount() {
    console.log('componentDidMount call');
  }

  componentDidUpdate() {
    console.log('componentDidUpdate call');
  }

  componentWillUpdate() {
    console.log('componentWillUpdate call');
  }

  componentWillUnmount() {
    console.log('componentWillUnmount call');
  }

  render() {
    console.log('render call');
    return (
      ...
    )
  }
}
```

```
cd 0_install/first_app
```

Run react server.

```
npm start
```

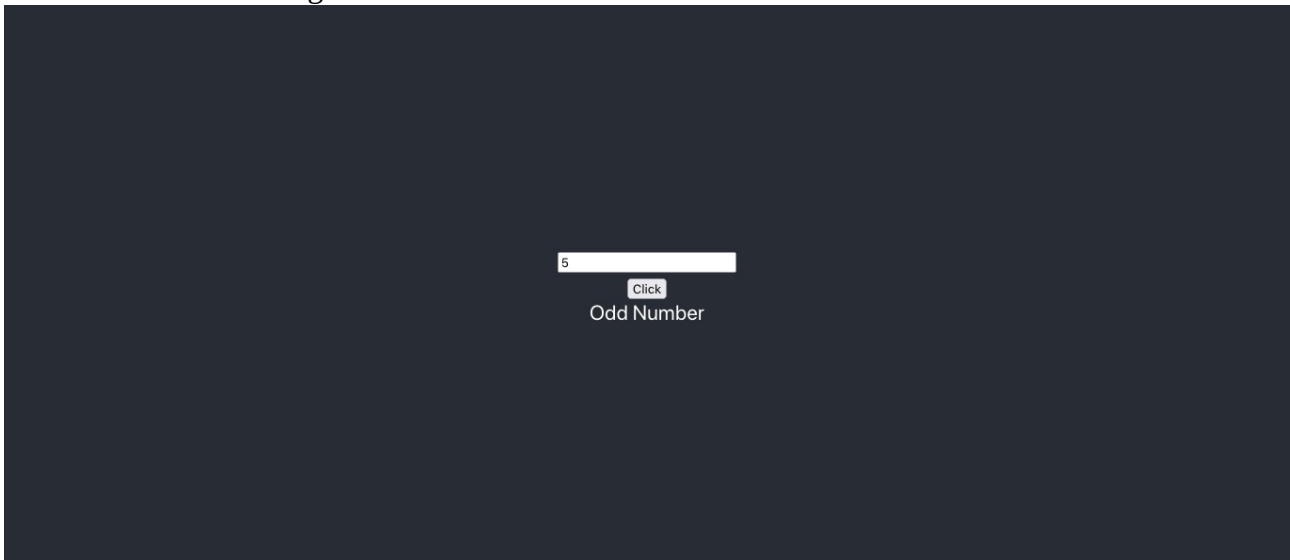
Create 6_life_cycle folder.

Copy 0_install/first_app/src folder to 6_life_cycle folder.

(7) React Conditional

Example 1

Even Odd Number using conditional.



Create component/even_odd.js file.

```
class EvenOdd extends Component {
  state = {
    num: 0
  }
  setNum = evt => {
    this.state.num = evt.target.value;
  }
  click = evt => {
    this.setState({ num: this.state.num });
  }
  render() {
    return (
      <div>
        <div>
          <input onChange={ this.setNum } type="text" />
        </div>
        <div>
          <button onClick={ this.click }> Click </button>
        </div>
        <div>
          { this.state.num % 2 == 0 ? "Even Number" : "Odd Number" }
        </div>
      </div>
    )
  }
}
export default EvenOdd;
```

Change App.js file.

```
import './App.css';
import EvenOdd from './components/even_odd.js';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <EvenOdd />
      </header>
    </div>
  );
}

export default App;
```

Example 2

Login and Logout screen using conditional.



A dark-themed login form is displayed on a black background. It consists of two labels, 'username:' and 'password:', each followed by a white rectangular input field. Below the password field is a small, light gray button with the text 'Login' in a dark font.

Copy even_odd.js and rename component/login.js file.

```

class Login extends Component {
  state = {
    isLogin: false,
    username: "",
  }
  setUser = evt => {
    this.state.username = evt.target.value;
  }
  clickLogin = evt => {
    if (this.state.username == 'kyaw' && this.state.passwd == '123') {
      this.setState({isLogin: true});
    }
  }
  clickLogout = evt => {
    this.setState({isLogin: false});
  }
  render() {
    return (
      <div>
        { this.state.isLogin ? (
          <div>
            <div>
              Welcome home screen !
            </div>
            <div>
              <button onClick={ this.clickLogout }> Logout </button>
            </div>
          </div>
        ) : (
          <div>
            <div>
              username: <input onChange={ this.setUser } type="text" />
            </div>
            <div>
              <button onClick={ this.clickLogin }> Login </button>
            </div>
          </div>
        )
      </div>
    )
  }
}
export default Login;

```

Create 7_conditional folder.

Copy 0_install/first_app/src folder to 7_conditional folder.

(8) React Keys

Create component/home.js file.

```
import React, { Component } from 'react';

class Home extends Component {
  render() {
    const employees = ['aung aung', 'mg mg', 'kyaw kyaw', 'aye aye']
    return (
      <div>
        { employees.map(employee => {
          return (
            <div>
              <h1> { employee } </h1>
            </div>
          )
        }) }
      </div>
    )
  }
}

export default Home;
```

Inspect browser console.

Warning: Each child in a list should have a unique "key" prop.

Check the render method of `Home`. See <https://reactjs.org/link/warning-keys> for more information.

div

Home@http://localhost:3000/static/js/bundle.js:354:1

header

div

App

Create component/new_home.js file.

```
import React, { Component } from 'react';

class NewHome extends Component {

  render() {
    const employees = [
      {id: 1, name: 'aung aung'},
      {id: 2, name: 'mg mg'},
      {id: 3, name: 'kyaw kyaw'},
      {id: 4, name: 'aye aye'}
    ]

    return (
      <div>
        { employees.map(employee => {
          return (
            <div key={employee.id}>
              <h1> { employee.name } </h1>
            </div>
          )
        })
        }
      </div>
    )
  }
}

export default NewHome;
```

aung aung

mg mg

kyaw kyaw

aye aye

Create 8_keys folder.

Copy 0_install/first_app/src folder to 8_keys folder.

(9) React Router

```
npm install react-router-dom
```

Check in package.json file.

```
npm list
```

Create component/menu.js file.

```
import { BrowserRouter, Routes, Route } from "react-router-dom";
import EvenOdd from './even_odd.js';
import Login from './login.js';
import Home from './new_home.js';

const Menu = () => {
  return (
    <BrowserRouter>
      <Routes>
        <Route path="login" element={<Login />} />
        <Route path="home" element={<Home />} />
        <Route path="evenodd" element={<EvenOdd />} />
      </Routes>
    </BrowserRouter>
  )
};

export default Menu;
```

Run in browser -

```
localhost:3000/login
localhost:3000/home
localhost:3000/evenodd
```

Create 9_router folder.

Copy 0_install/first_app/src folder to 9_router folder.

(10) React Hooks

useState

Copy component/login.js and Create component/new_login.js file.
Change class state instead function useState and remove this keyword.

```
import { useState } from 'react';

function NewLogin() {
  const [islogin, setlogin] = useState(false);
  const [username, setUsername] = useState("");

  const setUsername = evt => {
    setUsername(evt.target.value);
  }

  const clickLogin = evt => {
    if (username == 'kyaw' && passwd == '123') {
      setlogin(true);
    } else {
      alert('sorry, invalid username and password !');
    }
  }

  const clickLogout = evt => {
    setlogin(false);
  }
  ...
}

export default NewLogin;
```

Change Menu.js file.

```
...
import Login from './new_login.js';
...
```

useEffect

Change new_login.js file.

```
import { useState, useEffect } from 'react';

function NewLogin() {
  ...
  useEffect(() => {
    // run after every rendering
    if (islogin && username) {
      console.log(username, 'logged in.');
```

Add new users.

```
username = aung
passwd == 321

username = aye
passwd == 456
```

useContext

Copy component/menu.js and Create component/new_menu.js file.

```
import { BrowserRouter, Routes, Route } from "react-router-dom";
import EvenOdd from './even_odd.js';
import Login from './new_login.js';
import Home from './new_home.js';
import { createContext } from 'react';

export const Context = createContext();

const NewMenu = () => {

  const app_name = 'First App';

  return (
    <BrowserRouter>
      <Context.Provider value={app_name}>
        <Routes>
          ...
        </Routes>
      </Context.Provider>
    </BrowserRouter>
  )
};

export default NewMenu;
```

Change new_login.js file.

```
...
import { Context } from './new_menu.js';
...
return (
  <div>
    <Context.Consumer>
      { value => <span> { value } </span> }
    </Context.Consumer>
    { islogin ? (
  ...
```

Add Context Consumer in **new_home.js** and **even_odd.js** and app_name change to **Second App**.

Create 10_router folder.

Copy 0_install/first_app/src folder to 10_router folder.

(11) React Styling

CSS Stylesheet

Create src/Login.css file.

```
.Login {  
  background-color: blue;  
}
```

Change new_login.js file.

```
...  
import '../Login.css';  
...  
return (  
  <div className="Login">  
    <Context.Consumer>  
  </div>  
...)
```

Style Object

Change even_odd.js file.

```
...  
render() {  
  const myStyle = {  
    backgroundColor: "orange",  
  };  
  
  return (  
    <div style={myStyle}>  
      <Context.Consumer>  
    </div>  
  </div>  
...)
```

Inline Style

Change new_home.js file.

```
...  
return (  
  <div style={{ backgroundColor: "green" }}>  
    <Context.Consumer>  
  </div>  
...)
```

Create 11_style folder.

Copy 0_install/first_app/src folder to 11_style folder.

Django

1) Create new folder **4_django_restful** beside **3_react**.

```
mkdir 4_django_react
```

2) Check python version available 3.7.

```
py -0
```

3) Create new python virtual environment.

```
py -3.7 -m venv django2.2-venv
```

4) Activate virtual environment.

```
./django2.2-venv/Script/activate
```

5) Install django and check by pip list.

```
python -m pip install django==2.2
```

6) Create new project using django.

```
python -m django startproject hrms
```

7) Rename hrms project name as hrms-api.

```
cd hrms-api
```

8) Run server by manage.py file.

```
python manage.py runserver
```

9) Test in localhost:8000 in browser.

The install worked successfully! Congratulations!

Application Programming Interface (API)

1) Create new django application.

```
python manage.py startapp api
```

2) Register new app in setting.py

3) Database tables migrate.

```
python manage.py migrate
```

4) Check **hrms-api/db.sqlite3** database.

Download SQL Query Browser (www.sqlitebrowser.org)

5) Create new admin user.

```
python manage.py createsuperuser
```

username: admin

email: admin@gmail.com

password: superuser

6) Login django administration at localhost:8000/admin in browser.

7) Create new employee table.

hrms-api/api/models.py

```
from django.db import models
# Create your models here.
class EmployeeModel(models.Model):
    name = models.CharField(max_length=20)
    phone = models.CharField(max_length=20)
    address = models.CharField(max_length=20)
```

8) makemigrations and migrate for new change.

```
python manage.py makemigrations api
python manage.py migrate api
```

9) Register for django administration.

```
from django.contrib import admin
from .models import EmployeeModel

# Register your models here.
admin.site.register(EmployeeModel)
```


Django Restful Framework

1) Install django restful framework. (www.django-rest-framework.org)

```
python -m pip install djangorestframework==3.9.2
```

2) Register in setting.py

```
INSTALLED_APPS = [  
    ...  
    'django.contrib.staticfiles',  
    'rest_framework',  
    'api',  
]
```

3) Create new serializers.py

hrms-api/api/serializers.py

```
from rest_framework import serializers  
from .models import EmployeeModel  
  
class EmployeeSerializer(serializers.ModelSerializer):  
    class Meta:  
        model = EmployeeModel  
        fields = ['id', 'name', 'phone', 'address']
```

4) Edit views.py.

hrms-api/api/views.py

```
from django.shortcuts import render  
  
# Create your views here.  
from rest_framework import viewsets  
from .models import EmployeeModel  
from .serializers import EmployeeSerializer  
  
class EmployeeViewSet(viewsets.ModelViewSet):  
    serializer_class = EmployeeSerializer  
    queryset = EmployeeModel.objects.all()
```

5) Creat new urls.py

hrms-api/api/urls.py

```
from rest_framework import routers
from django.urls import path, include
from .views import EmployeeViewSet

router = routers.DefaultRouter()
router.register('employees', EmployeeViewSet)

urlpatterns = [
    path("", include(router.urls))
]
```

6) Edit root urls.py

hrms-api/hrms/urls.py

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('api/', include('api.urls'))
]
```

7) Run **localhost:8000/api/employees** in browser.

Test API Method

POST (Create new employee)

```
HTTP 201 Created
Allow: GET, POST, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept

{
  "id": 2,
  "name": "Mg Mg",
  "phone": "09787897878",
  "address": "Mandalay"
}
```

GET (Read employee)

HTTP 200 OK

Allow: GET, POST, HEAD, OPTIONS

Content-Type: application/json

Vary: Accept

```
[
  {
    "id": 1,
    "name": "Kyaw Kyaw",
    "phone": "09383838",
    "address": "Yangon"
  }
]
```

PUT (Update employee)

Change url => **http://localhost:8000/api/employees/2**

HTTP 200 OK

Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS

Content-Type: application/json

Vary: Accept

```
{
  "id": 2,
  "name": "Maung Maung",
  "phone": "09787897878",
  "address": "Mandalay"
}
```

DELETE (Delete employee)

Change url => **http://localhost:8000/api/employees/2**

HTTP 204 No Content

Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS

Content-Type: application/json

Vary: Accept

Are you sure you want to delete this Employee Instance?

Auth Token

1) Register in setting.py

```
INSTALLED_APPS = [  
    ...  
    'rest_framework',  
    'rest_framework.authtoken',  
    'api',  
]
```

2) Migrate auth token table.

```
Python manage.py migrate
```

3) Create token for admin user at django administration.

```
5bfc020cdc3bbe1f3e399fe2c5727c6c7e85c28a
```

Postman API Platform

Download postman (www.postman.com)

1) Edit root urls.py

hrms-api/hrms/urls.py

```
from django.contrib import admin  
from django.urls import path, include  
from rest_framework.authtoken.views import obtain_auth_token  
  
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path('api/', include('api.urls')),  
    path('auth/', obtain_auth_token)  
]
```

2) In postman api platform.

POST => **localhost:8000/auth/**.

Body Form Data

```
username = admin  
password = superuser
```

Return Result

```
{"token": "5bfc020cdc3bbe1f3e399fe2c5727c6c7e85c28a"}
```

Permission IsAuthenticated

1) Edit setting.py

```
...
WSGI_APPLICATION = 'hrms.wsgi.application'

REST_FRAMEWORK = {
    'DEFAULT_PERMISSION_CLASSES': (
        'rest_framework.permissions.IsAuthenticated',
    )
}
...
```

2) Edit views.py

```
from django.shortcuts import render

# Create your views here.
from rest_framework import viewsets
from .models import EmployeeModel
from .serializers import EmployeeSerializer
from rest_framework.authentication import TokenAuthentication

class EmployeeViewSet(viewsets.ModelViewSet):
    serializer_class = EmployeeSerializer
    queryset = EmployeeModel.objects.all()
    authentication_classes = (TokenAuthentication,)
```

3) In postman api platform.

GET => **localhost:8000/employees.**

Body Form Data

username = admin
password = superuser

Return Result

```
{
  "detail": "Authentication credentials were not provided."
}
```

4) Include headers.

GET => **localhost:8000/employees.**

Body Form Data

username = admin

password = superuser

Headers

Authorization = Token 5bfc020cdc3bbe1f3e399fe2c5727c6c7e85c28a

Return Result

```
[
  {
    "id": 1,
    "name": "Kyaw Kyaw Kwa",
    "phone": "09383838",
    "address": "Yangon"
  }
]
```

Django React

Inside **4_django_restful** folder.

```
npx create-react-app hrms-web
```

project structure

- hrms-web -
 - node_modules
 - package.json
 - package-lock.json
 - public
 - **src** -
 - App.css
 - App.js
 - index.css
 - index.js
 - reportWebVitals.js
 - **components** -
 - home.js
 - login.js
 - menu.js

home.js

```
import React, { Component } from 'react';

class Home extends Component {
  render() {
    const employees = ['aung aung', 'mg mg', 'kyaw kyaw', 'aye aye']
    return (
      <div>
        { employees.map(employee => {
          return (
            <div>
              <h1> { employee } </h1>
            </div>
          )
        })
        }
      </div>
    )
  }
}

export default Home;
```

menu.js

```
import { BrowserRouter, Routes, Route } from "react-router-dom";
import Login from './login.js';
import Home from './home.js';

const Menu = () => {
  return (
    <BrowserRouter>
      <Routes>
        <Route path="login" element={ <Login /> } />
        <Route path="home" element={ <Home /> } />
      </Routes>
    </BrowserRouter>
  )
};

export default Menu;
```

App.js

```
import './App.css';
import Menu from './components/menu.js';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <Menu />
      </header>
    </div>
  );
}

export default App;
```

Test url

localhost:3000/home

localhost:3000/login

(1) Using Fetch Method

home.js

```
import React, { Component } from 'react';

class Home extends Component {

  componentDidMount(){
    fetch('http://127.0.0.1:8000/api/employees/', {
      method: 'GET',
      headers: {
        'Content-Type': 'application/json',
        'Authorization': 'Token 5bfc020cdc3bbe1f3e399fe2c5727c6c7e85c28a'
      }
    })
    .then( resp => resp.json())
    .then( res => console.log(res))
    .catch( error => console.log(error))
  }

  render() {
    const employees = ['aung aung', 'mg mg', 'kyaw kyaw', 'aye aye']
    return (
      <div>
        { employees.map(employee => {
          return (
            <div>
              <h1> { employee } </h1>
            </div>
          )
        }) }
      </div>
    )
  }
}

export default Home;
```

Key Error

Warning: Each child in a list should have a unique "key" prop.

Update home.js

```
render() {
  const employees = ['aung aung', 'mg mg', 'kyaw kyaw', 'aye aye']
  return (
    <div>
      { employees.map(employee => {
        return (
          <div key= { employee }>
            <h1> { employee } </h1>
          </div>
        )
      })
    }
  </div>
  )
}
```

Cross-Origin Request Blocked

Cross-Origin Request Blocked: The Same Origin Policy disallows reading the remote resource at <http://127.0.0.1:8000/api/employees/>. (Reason: CORS header 'Access-Control-Allow-Origin' missing). Status code: 401.

Install django-cors-headers

```
python -m pip install django-cors-headers==2.5.3
```

Change hrms-api/hrms/settings.py

```
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'corsheaders.middleware.CorsMiddleware',
    ...
]

CORS_ORIGIN_WHITELIST = (
    'localhost:3000'
)
```

Update return data into state

Change home.js

```
import React, { Component } from 'react';

class Home extends Component {

  state = {
    employees: []
  }

  componentDidMount(){
    fetch('http://127.0.0.1:8000/api/employees/', {
      method: 'GET',
      headers: {
        'Content-Type': 'application/json',
        'Authorization': 'Token 5bfc020cdc3bbe1f3e399fe2c5727c6c7e85c28a'
      }
    })
    .then( resp => resp.json())
    .then( res => this.setState({ employees: res }))
    .catch( error => console.log(error))
  }

  render() {
    //const employees = ['aung aung', 'mg mg', 'kyaw kyaw', 'aye aye']
    return (
      <div>
        {
          this.state.employees.map(employee => {
            return (
              <div key={ employee.id }>
                <h1> { employee.name } </h1>
              </div>
            )
          })
        }
      </div>
    )
  }
}

export default Home;
```

(2) List View

Create hrms-web/src/components/employees folder.
Create detail.js, list.js and form.js

project structure

```
- hrms-web -  
  ...  
  - public  
  - src -  
    - App.css  
    - App.js  
    ...  
    - components -  
      - home.js  
      - login.js  
      - menu.js  
      - employees  
        - detail.js  
        - list.js  
        - form.js
```

Change list.js

```
import React, { Component } from 'react';  
  
class EmployeeList extends Component {  
  render() {  
    return (  
      <div>  
        <h1> List View </h1>  
        {  
          this.props.employees.map(employee => {  
            return (  
              <div key={ employee.id }>  
                <h3> { employee.name } </h3>  
              </div>  
            )  
          })  
        }  
      </div>  
    )  
  }  
}  
  
export default EmployeeList;
```

Change home.js

```
import React, { Component } from 'react';
import EmployeeList from './employees/list.js';

class Home extends Component {

  state = {
    employees: []
  }

  componentDidMount(){
    ...
  }

  render() {
    return (
      <div>
        <h1> HRMS </h1>
        <div>
          <EmployeeList employees={ this.state.employees }/>
        </div>
      </div>
    )
  }
}

export default Home;
```

HRMS
List View
Kyaw Kyaw
Aung Aung

(3) Detail View

detail.js

```
import React, { Component } from 'react';

class EmployeeDetail extends Component {

    render() {
        return (
            <div>
                <h1> Detail View </h1>
            </div>
        )
    }
}

export default EmployeeDetail;
```

home.js

```
import React, { Component } from 'react';
import EmployeeList from './employees/list.js';
import EmployeeDetail from './employees/detail.js';

class Home extends Component {

    state = {
        employees: []
    }

    componentDidMount(){
        ...
    }

    render() {
        return (
            <div>
                <h1> HRMS </h1>
                <div>
                    <EmployeeList employees={ this.state.employees }/>
                    <EmployeeDetail />
                </div>
            </div>
        )
    }
}

export default Home;
```

Create src/Home.css

Home.css

```
.Header {  
  display: grid;  
  grid-template-columns: 1fr 1fr;  
  text-align: left;  
  grid-gap: 100px;  
}
```

Import css at home.js

```
import '../Home.css';  
  
<h1> HRMS </h1>  
  <div className="Header">
```



(3) List Item Click

home.js

```
import React, { Component } from 'react';
import EmployeeList from './employees/list.js';
import EmployeeDetail from './employees/detail.js';
import '../Home.css';

class Home extends Component {

  state = {
    employees: []
  }

  componentDidMount(){
    ...
  }

  employeeClicked = employee => {
    console.log("Employee Clicked ", employee);
  }

  render() {
    return (
      <div>
        <h1> HRMS </h1>
        <div className="Header">
          <EmployeeList employees={this.state.employees}
            employeeClicked={this.employeeClicked}/>
          <EmployeeDetail />
        </div>
      </div>
    )
  }
}

export default Home;
```


list.js

```
import React, { Component } from 'react';

class EmployeeList extends Component {

  employeeClicked = employee => evt => {
    this.props.employeeClicked(employee);
  };

  render() {
    return (
      <div>
        <h1> Employee List </h1>
        {
          his.props.employees.map(employee => {
            return (
              <div key={ employee.id }>
                <h3
                  onClick={this.employeeClicked(employee)}
```

console result

```
Employee Clicked
Object { id: 1, name: "Kyaw Kyaw", phone: "09383838", address: "Yangon" }
home.js:28

Employee Clicked
Object { id: 4, name: "Aung Aung", phone: "09373737373", address: "Mandalay" }
home.js:28
```

(4) Connect List View with Detail View

home.js

```
state = {
  employees: [],
  selectedEmployee: null
}

employeeClicked = employee => {
  console.log("Employee Clicked ", employee);
  this.setState({ selectedEmployee: employee });
}

render() {
  return (
    ...
    <EmployeeDetail employee={ this.state.selectedEmployee } />
    ...
  )
}
```

detail.js

```
import React, { Component } from 'react';

class EmployeeDetail extends Component {

  render() {

    const employee = this.props.employee;

    return (
      <div>
        <h1> Detail View </h1>
        <div>
          <h3> {employee.name} </h3>
          <p> {employee.phone} </p>
          <p> {employee.address} </p>
        </div>
      </div>
    )
  }
}

export default EmployeeDetail;
```

Uncaught TypeError: employee is null

Update home.js

```
{
  this.state.selectedEmployee
  ?
    <EmployeeDetail employee={ this.state.selectedEmployee } />
  :
    null
}
```



(5) Form View

Change form.js

```
import React, { Component } from 'react';

class EmployeeForm extends Component {
  render() {
    return (
      <div>
        <h1> Form View </h1>
      </div>
    )
  }
}

export default EmployeeForm;
```

Change home.js

Add new state view type and add new button.

```
...
import EmployeeForm from './employees/form.js';

class Home extends Component {
  state = {
    ...
    view_type: ''
  }

  componentDidMount(){
    ...
  }

  employeeClicked = employee => {
    ...
    this.setState({ selectedEmployee: employee, view_type: 'detail' });
  }

  addNewClicked = () => {
    console.log("Add New Click");
    this.setState({ view_type: 'create' });
  };

  render() {
    return (
      <div>
        ...
        <div className="Header">
          {
            this.state.view_type == 'create'
            ? <EmployeeForm/> : null
          }

          {
            this.state.view_type == 'detail'
            ? <EmployeeDetail
              employee={ this.state.selectedEmployee } />
            : null
          }
        </div>
        <div className="Footer">
          <button onClick={this.addNewClicked}>Add New</button>
        </div>
      </div>
    )
  }
}
```

Add new css class at Home.css

```
.Footer {  
  text-align: left;  
}
```

Change form.js

```
class EmployeeForm extends Component {  
  save = () => {  
    console.log('save click');  
  }  
  
  render() {  
    return (  
      <div>  
        <h1> Form View </h1>  
        <div>  
          <span> Name </span> <br/>  
          <input name="name" type="text" /> <br/>  
          <span> Phone </span> <br/>  
          <input name="phone" type="text" /> <br/>  
          <span> Address </span> <br/>  
          <input name="address" type="text" /> <br/>  
          <button onClick={this.save}> Save </button>  
        </div>  
      </div>  
    )  
  }  
}
```

HRMS

Employee List

Kyaw Kyaw

Aung Aung

Add New

Form View

Name

Phone

Address

Save

(6) Form Save

Change form.js

```
import React, { Component } from 'react';

class EmployeeForm extends Component {

  state = {
    editedEmployee: { 'name': '', 'phone': '', 'address': '' }
  }

  inputChanged = evt => {
    console.log('input change', evt.target.value);
    let employee = this.state.editedEmployee;
    employee[evt.target.name] = evt.target.value;
    this.setState({ editedEmployee: employee });
  }

  save = () => {
    console.log('save click');
    fetch('http://127.0.0.1:8000/api/employees/', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
        'Authorization': 'Token 5bfc020cdc3bbe1f3e399fe2c5727c6c7e85c28a'
      },
      body: JSON.stringify(this.state.editedEmployee)
    }).then( resp => resp.json())
    .then( res => console.log(res))
    .catch( error => console.log(error))
  }

  render() {
    return (
      <div>
        <h1> Form View </h1>
        <div>
          <span> Name </span> <br/>
          <input name="name" type="text" onChange={this.inputChanged}/> <br/>
          ...
        </div>
      </div>
    )
  }
}

export default EmployeeForm;
```

(7) Form Update

Font Awesome Install

```
npm install react-fontawesome
```

public/index.html

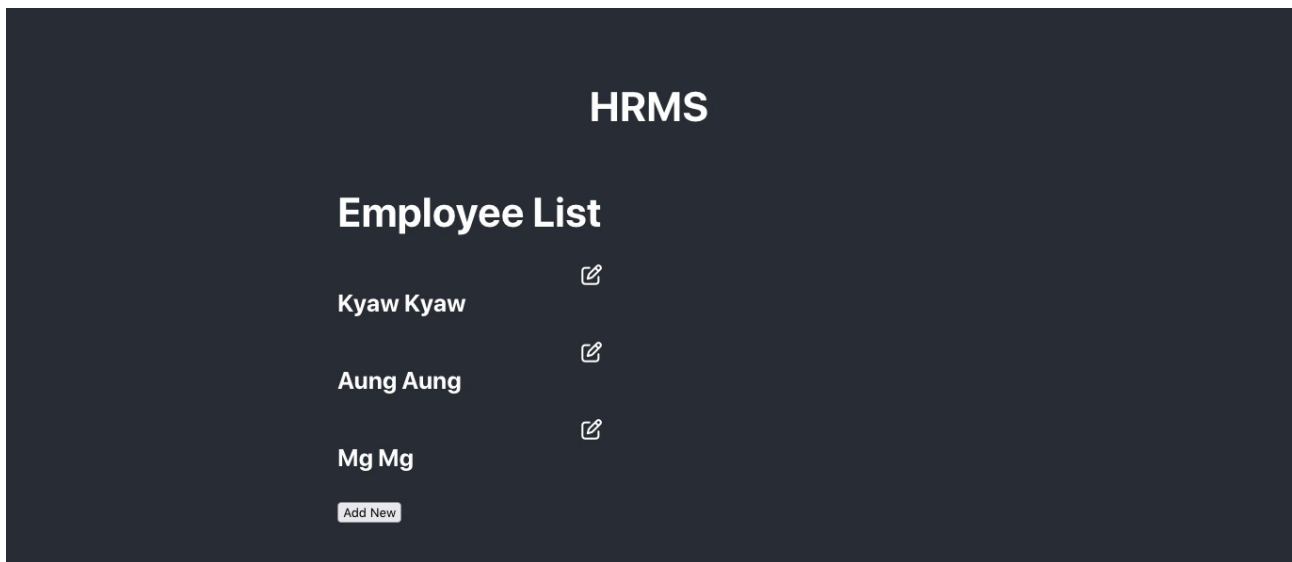
```
<script src="https://kit.fontawesome.com/7f0377bcc2.js" crossorigin="anonymous"></script>
```

List.css

```
.List {  
  display: grid;  
  grid-template-columns: 1fr auto auto;  
}
```

list.js

```
import React, { Component } from 'react';  
import './List.css';  
var FontAwesome = require('react-fontawesome');  
  
class EmployeeList extends Component {  
  ...  
  
  updateClicked = employee => evt => {  
    this.props.updateClicked(employee);  
  };  
  
  render() {  
    return (  
      <div>  
        <h1> Employee List </h1>  
        {  
          this.props.employees.map(employee => {  
            return (  
              <div key={employee.id} className="List">  
                ...  
                <FontAwesome name="edit"  
                  onClick={this.updateClicked(employee)}</b>  
              </div>  
            )  
          })  
        }  
      </div>  
    )  
  }  
  ...  
}
```



Change home.js

```

addNewClicked = () => {
  console.log("Add New Click");
  let newemployee = {name: '', 'phone': '', 'address': ''};
  this.setState({selectedEmployee: newemployee, view_type: 'create'});
};

updateClicked = employee => {
  console.log("update Click");
  this.setState({selectedEmployee: employee, view_type: 'update'});
};

render() {
  return (
    <div>
      <h1> HRMS </h1>
      <div className="Header">
        <EmployeeList employees={this.state.employees}
          employeeClicked={this.employeeClicked}
          updateClicked={this.updateClicked}/>
        {
          this.state.view_type == 'create'
          || this.state.view_type == 'update'
          ?
            <EmployeeForm . . ./>
          :
            null
        }
      </div>
      . . .
    </div>
  )
  . . .
}

```



```

import React, { Component } from 'react';

class EmployeeForm extends Component {

  state = {
    editedEmployee: null
  }

  inputChanged = evt => {
    console.log('input change', evt.target.value);
    let employee = this.props.employee;
    employee[evt.target.name] = evt.target.value;
    this.setState({ editedEmployee: employee });
  }

  save = () => {
    ...
  }

  update = () => {
    console.log('update click');
    fetch(`http://127.0.0.1:8000/api/employees/${this.props.employee.id}/`, {
      method: 'PUT',
      headers: {
        'Content-Type': 'application/json',
        'Authorization': 'Token 5bfc020cdc3bbe1f3e399fe2c5727c6c7e85c28a'
      },
      body: JSON.stringify(
        this.state.editedEmployee
      )
    }).then( resp => resp.json())
    .then( res => console.log(res))
    .catch( error => console.log(error))
  }

  render() {
    return (
      <div>
        <h1> Form View </h1>
        <div>
          <input name="name" type="text" value={this.props.employee.name}
            onChange={this.inputChanged}/> <br/>
          ...
          <button onClick={this.save}> Save </button>
          <button onClick={this.update}> Update </button>
        </div>
      </div>
    )
  }
}

```

(8) Delete

list.js

```
...

deleteClicked = employee => evt => {
  console.log('delete click');
  fetch(`http://127.0.0.1:8000/api/employees/${employee.id}/`, {
    method: 'DELETE',
    headers: {
      'Content-Type': 'application/json',
      'Authorization': 'Token 5bfc020cdc3bbe1f3e399fe2c5727c6c7e85c28a'
    }
  })
  .then( res => this.props.reload(employee, 'delete'))
  .catch( error => console.log(error))
};

...

<FontAwesome name="edit" onClick={this.updateClicked(employee)}/>
<FontAwesome name="trash" onClick={this.deleteClicked(employee)}/>
...
```

(9) Reload

home.js

```
reload = (employee, method) => {
  if (method === 'save') {
    this.setState({employees: [...this.state.employees, employee]});
  } else if (method === 'update') {
    this.setState({view_type: ""});
  } else if (method === 'delete') {
    const employees = this.state.employees.filter( emp => emp.id !==
      employee.id);
    this.setState({employees: employees});
  }
}
```