

## ДОДАТОК А

## Код програми

```

    import React from 'react';
    import ReactDOM from 'react-dom';
    import App from './App';

    ReactDOM.render(
      <App />,
      document.getElementById('root')
    );

    import React from 'react'
    import {BrowserRouter as Router, Route} from 'react-router-dom'
    import "bootstrap/dist/css/bootstrap.min.css"

    import Navbar from './components/navbar.component'
    import MainList from './components/main-list.component'
    import WorkertList from "./components/student-list.component";
    import ProcessList from "./components/quiz-list.component";
    import DutyList from "./components/duty-list.component";
    import PayList from "./components/pay-list.component";

    function App() {

      return (
        <div>
          <Router>
            <div className='container'>
              <Navbar/>
              <br/>
              <Route path='/news' exact component={MainList}/>
              <Route path='/pays' exact component={PayList}/>
              <Route path='/students' component={WorkertList}/>
              <Route path='/quizzes' component={ProcessList}/>
              <Route path='/dutys' component={DutyList}/>
            </div>
          </Router>
        </div>
      );
    }

    export default App;

    const express = require('express')
    const cors = require('cors')
    const mongoose = require('mongoose')

    require('dotenv').config()

    const app = express()
    const port = process.env.PORT || 5000

```

```

app.use(cors())
app.use(express.json())

const uri = process.env.ATLAS_URI
mongoose.connect(uri, {useNewUrlParser: true, useCreateIndex: true, useUnifiedTopology: true})
const connection = mongoose.connection
connection.once('open', () => {
  console.log('MongoDB database connection established successfully')
})
//Computed
import * from 'models'
let dessionion{
this.processworkers = processdescription / (processplan * workereffect )
this.paysum = (fullTimeIncrement(date = new Date) + dutyTimeIncrement(date = new Date)) 30 //for
example stavka
}
module.exports = dessionion
{
  "name": "diplom_work",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@testing-library/jest-dom": "^5.11.6",
    "@testing-library/react": "^11.2.2",
    "@testing-library/user-event": "^12.2.2",
    "axios": "^0.21.0",
    "bootstrap": "^4.5.3",
    "cors": "^2.8.5",
    "dotenv": "^8.2.0",
    "express": "^4.17.1",
    "mongoose": "^5.10.15",
    "react": "^17.0.1",
    "react-dom": "^17.0.1",
    "react-router-dom": "^5.2.0",
    "react-scripts": "4.0.1",
    "web-vitals": "^0.2.4"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",

```

```

    "last 1 firefox version",
    "last 1 safari version"
  ]
}
}
const mainRouter = require('./routes/main')
app.use('/main', mainRouter)
const studentRouter = require('./routes/worker')
app.use('/student', workerRouter)
const quizRouter = require('./routes/process')
app.use('/quiz', processRouter)
const dutyRouter = require('./routes/duty')
app.use('/duty', dutyRouter)
const payRouter = require('./routes/pay')
app.use('/pay', payRouter)

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

const mongoose = require('mongoose')

const Schema = mongoose.Schema

const dutySchema = new Schema({
  dutyname: {type: String, required: true},
  dutylastname: {type: String, required: true},
  dutydate: {type: String, required: true},
},
{
  timestamps: true
})

const Duty = mongoose.model('Duty', dutySchema)

module.exports = Duty

const mongoose = require('mongoose')

const Schema = mongoose.Schema

const mainSchema = new Schema({
  posttitle: {type: String, required: true},
  postdescription: {type: String, required: true},
},
{
  timestamps: true
})

const Main = mongoose.model('Main', mainSchema)

module.exports = Main

const mongoose = require('mongoose')

const Schema = mongoose.Schema

```

```

const paySchema = new Schema({
  payname: {type: String, required: true},
  paylastname: {type: String, required: true},
  paysum: {type: String, required: true},
},
{
  timestamps: true
})

const Pay = mongoose.model('Pay', paySchema)

module.exports = Pay

const mongoose = require('mongoose')

const Schema = mongoose.Schema

const quizSchema = new Schema({
  quiztitle: {type: String, required: true},
  quizdescription: {type: String, required: true},
  quizyes: {type: Number, required: true},
  quizno: {type: Number, required: true},
},
{
  timestamps: true
})

const mongoose = require('mongoose')

const Schema = mongoose.Schema

const processSchema = new Schema({
  processtitle: {type: String, required: true},
  processdescription: {type: String, required: true},
  processyes: {type: Number, required: true},
  processno: {type: Number, required: true},
},
{
  timestamps: true
})

const Process = mongoose.model('Process', processSchema)

module.exports = Process

const Worker = mongoose.model('Student', studentSchema)

module.exports = Worker

const router = require('express').Router()
let Duty = require('../models/duty.model')

router.route('/').get((req, res) => {
  Duty.find()
  .then(duty => res.json(duty))
})

```

```

        .catch(err => res.status(400).json('Error' + err))
    })

    router.route('/add').post((req, res) => {
        const dutyname = req.body.dutyname;
        const dutylastname = req.body.dutylastname;
        const dutydate = req.body.dutydate;

        const newDuty = new Duty({
            dutyname,
            dutylastname,
            dutydate
        })

        newDuty.save()
            .then(() => res.json('Duty added'))
            .catch(err => res.status(400).json('Error' + err))
    })

    router.route('/:id').get((req, res) => {
        Duty.findById(req.params.id)
            .then(duty => res.json(duty))
            .catch(err => res.status(400).json('Error' + err))
    })

    router.route('/:id').delete((req, res) => {
        Duty.findByIdAndDelete(req.params.id)
            .then(() => res.json('Duty deleted!'))
            .catch(err => res.status(400).json('Error' + err))
    })

    module.exports = router

    const router = require('express').Router()
    let Main = require('../models/main.model')

    router.route('/').get((req, res) => {
        Main.find()
            .then(main => res.json(main))
            .catch(err => res.status(400).json('Error' + err))
    })

    router.route('/add').post((req, res) => {
        const posttitle = req.body.posttitle;
        const postdescription = req.body.postdescription;

        const newMain = new Main({
            posttitle,
            postdescription,
        })

        newMain.save()
            .then(() => res.json('Post added'))
            .catch(err => res.status(400).json('Error' + err))
    })

```

```
router.route('/:id').get((req, res) => {
  Main.findById(req.params.id)
  .then(main => res.json(main))
  .catch(err => res.status(400).json('Error' + err))
})
```

```
router.route('/:id').delete((req, res) => {
  Main.findByIdAndDelete(req.params.id)
  .then(() => res.json('Post deleted!'))
  .catch(err => res.status(400).json('Error' + err))
})
```

```
module.exports = router
```

```
const router = require('express').Router()
let Pay = require('../models/pay.model')
```

```
router.route('/').get((req, res) => {
  Pay.find()
  .then(pay => res.json(pay))
  .catch(err => res.status(400).json('Error' + err))
})
```

```
router.route('/add').post((req, res) => {
  const payname = req.body.payname;
  const paylastname = req.body.paylastname;
  const paysum = req.body.paysum;
```

```
  const newPay = new Pay({
    payname,
    paylastname,
    paysum
  })
```

```
  newPay.save()
  .then(() => res.json('Pay added'))
  .catch(err => res.status(400).json('Error' + err))
})
```

```
router.route('/:id').get((req, res) => {
  Pay.findById(req.params.id)
  .then(pay => res.json(pay))
  .catch(err => res.status(400).json('Error' + err))
})
```

```
router.route('/:id').delete((req, res) => {
  Pay.findByIdAndDelete(req.params.id)
  .then(() => res.json('Pay deleted!'))
  .catch(err => res.status(400).json('Error' + err))
})
```

```
module.exports = router
```

```
const router = require('express').Router()
let Process = require('../models/process.model')
```

```

router.route('/').get((req, res) => {
  Process.find()
    .then(process => res.json(process))
    .catch(err => res.status(400).json('Error' + err))
})

router.route('/add').post((req, res) => {
  const processtitle = req.body.processtitle;
  const processdescription = req.body.processdescription;
  const processplan = req.body.processplan;
  const processworkers = req.body.processworkers;

  const newProcess = new Process({
    processtitle,
    processdescription,
    processplan,
    processworkers
  })

  newProcess.save()
    .then(() => res.json('Process added'))
    .catch(err => res.status(400).json('Error' + err))
})

router.route('/:id').get((req, res) => {
  Process.findById(req.params.id)
    .then(process => res.json(process))
    .catch(err => res.status(400).json('Error' + err))
})

router.route('/:id').delete((req, res) => {
  Process.findByIdAndDelete(req.params.id)
    .then(() => res.json('Process deleted!'))
    .catch(err => res.status(400).json('Error' + err))
})

router.route('/update/:id').post((req, res) => {
  Process.findById(req.params.id)
    .then(process => {
      process.processplan = req.body.processplan;
      process.processworkers = req.body.processworkers;

      process.save()
        .then(() => res.json('Process updated!'))
        .catch(err => res.status(400).json('Error: ' + err));
    })
    .catch(err => res.status(400).json('Error: ' + err));
});

module.exports = router

const router = require('express').Router()
let Worker = require('../models/worker.model')

router.route('/').get((req, res) => {

```

```

Worker.find()
  .then(worker => res.json(worker))
  .catch(err => res.status(400).json('Error' + err))
})

router.route('/add').post((req, res) => {
  const workername = req.body.worername;
  const workerlastname = req.body.workerlastname;
  const workerprof = req.body.workerprof;
  const workereffect = req.body.workereffect;

  const newWorker = new Worker({
    workername,
    workerlastname,
    workerprof,
    workereffect
  })

  newWorker.save()
    .then(() => res.json('Worker added'))
    .catch(err => res.status(400).json('Error' + err))
})

router.route('/:id').get((req, res) => {
  Worker.findById(req.params.id)
    .then(worker => res.json(worker))
    .catch(err => res.status(400).json('Error' + err))
})

router.route('/:id').delete((req, res) => {
  Worker.findByIdAndDelete(req.params.id)
    .then(() => res.json('Worker deleted!'))
    .catch(err => res.status(400).json('Error' + err))
})

module.exports = router

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

const Duty = props => (
  <tr>
    <td>{props.duty.dutyname}</td>
    <td>{props.duty.dutylastname}</td>
    <td>{props.duty.dutydate}</td>
    <td>
      <a href="#" onClick={() => {
        props.deleteDuty(props.duty._id)
      }}>
        <button type="submit" className="btn btn-primary" data-toggle="button" aria-pressed="false">
          Відчегував
        </button>
      </a>
    </td>
  </tr>
)

```

```

export default class DutyList extends Component {
  constructor(props) {
    super(props);

    this.deleteDuty = this.deleteDuty.bind(this)

    this.state = {
      dutyname: "",
      dutylastname: "",
      dutydate: "",
      dutys: [],
    };
  }

  componentDidMount() {
    axios.get('http://localhost:5000/duty/')
      .then(response => {
        this.setState({ dutys: response.data })
      })
      .catch((error) => {
        console.log(error);
      })
  }

  deleteDuty(id) {
    axios.delete('http://localhost:5000/duty/' + id)
      .then(response => {
        console.log(response.data)
      });

    this.setState({
      dutys: this.state.dutys.filter(el => el._id !== id)
    })
  }

  dutyList() {
    return this.state.dutys.map(currentduty => {
      return <Duty duty={currentduty} deleteDuty={this.deleteDuty}
        key={currentduty._id}/>;
    })
  }

  render() {
    return (
      <div>
        <h3>Чергові</h3>
        <table className="table">
          <thead className="thead-light">
            <tr>
              <th>Ім'я</th>
              <th>Прізвище</th>
              <th>Дата чергування</th>
            </tr>
          </thead>
          <tbody>

```

```

        {this.dutyList()}
      </tbody>
    </table>
  </div>
)
}
}

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

const Main = props => (
  <div className="card">
    <div className="card-header">
      {props.main.posttitle}
    </div>
    <div className="card-body">
      <h5 className="card-title">{props.main.posttitle}</h5>
      <p className="card-text">{props.main.postdescription}</p>
      <a href="#" onClick={() => {
        props.deleteBike(props.main._id)
      }}>
        <button type="submit" className="btn btn-primary" data-toggle="button" aria-pressed="false">
          Видалити
        </button>
      </a>
    </div>
  </div>
)

export default class MainList extends Component {
  constructor(props) {
    super(props);

    this.deleteMain = this.deleteMain.bind(this)
    this.onChangePosttitle = this.onChangePosttitle.bind(this);
    this.onChangePostdescription = this.onChangePostdescription.bind(this);
    this.onSubmit = this.onSubmit.bind(this)

    this.state = {
      posttitle: "",
      postdescription: "",
      mains: [],
    };
  }

  componentDidMount() {
    axios.get("http://localhost:5000/main/")
      .then(response => {
        this.setState({ mains: response.data })
      })
      .catch((error) => {
        console.log(error);
      })
  }
}

```

```

deleteMain(id) {
  axios.delete('http://localhost:5000/main/' + id)
    .then(response => {
      console.log(response.data)
    });

  this.setState({
    mains: this.state.mains.filter(el => el._id !== id)
  })
}

onChangePosttitle(e) {
  this.setState({
    posttitle: e.target.value
  })
}

onChangePostdescription(e) {
  this.setState({
    postdescription: e.target.value
  })
}

onSubmit(e) {
  e.preventDefault();

  const main = {
    posttitle: this.state.posttitle,
    postdescription: this.state.postdescription
  }

  console.log(main);

  axios.post('http://localhost:5000/main/add', main)
    .then(res => console.log(res.data));

  window.location = '/';
}

mainList() {
  return this.state.mains.map(currentmain => {
    return <Main main={currentmain} deleteBike={this.deleteMain}
      key={currentmain._id}/>;
  })
}

render() {
  return (
    <div>

      <div>
        <h3>Додати новину</h3>
        <form onSubmit={this.onSubmit}>
          <div className="form-row">
            <div className="form-group col-md-3">
              <label>Заголовок</label>

```

```

        <input type="text"
            required
            value={ this.state.posttitle }
            onChange={ this.onChangePosttitle }
            className="form-control"/>
    </div>
    <div className="form-group col-md-7">
        <label>Новина</label>
        <textarea type="text"
            required
            value={ this.state.postdescription }
            onChange={ this.onChangePostdescription }
            className="form-control"/>
    </div>
    <div>
        <button type="submit" className="btn btn-primary" data-toggle="button" aria-
pressed="false"
            style={{ margin: 30 }}>
            Опублікувати
        </button>
    </div>
</div>

</form>
</div>
<h3>Новини</h3>
{this.mainList()}
<br/>
</div>
)
}
}

```

```

import React, {Component} from 'react'
import {Link} from "react-router-dom";

```

```

export default class Navbar extends Component {
    render() {
        return (
            <nav className="navbar navbar-dark bg-dark navbar-expand-lg">
                <Link to="/" className="navbar-brand">LOGO</Link>
                <div className="collapse navbar-collapse">
                    <ul className="navbar-nav mr-auto">
                        <li className="navbar-item">
                            <Link to="/news" className="nav-link">Новини</Link>
                        </li>
                        <li className="navbar-item">
                            <Link to="/pays" className="nav-link">Заробітна плата</Link>
                        </li>
                        <li className="navbar-item">
                            <Link to="/workers" className="nav-link">Працівники </Link>
                        </li>
                        <li className="navbar-item">
                            <Link to="/process" className="nav-link">Робочі процеси</Link>
                        </li>
                        <li className="navbar-item">
                            <Link to="/dutys" className="nav-link">Додаткові зміни</Link>

```

```

        </li>
    </ul>
</div>
</nav>

    )
}
}

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

const Pay = props => (
    <tr>
        <td>{props.pay.payname}</td>
        <td>{props.pay.paylastname}</td>
        <td>{props.pay.paysum}</td>
    </tr>
)

export default class PayList extends Component {
    constructor(props) {
        super(props);

        this.state = {
            payname: "",
            paylastname: "",
            paysum: "",
            pays: [],
        };
    }

    componentDidMount() {
        axios.get('http://localhost:5000/pay/')
            .then(response => {
                this.setState({pays: response.data})
            })
            .catch((error) => {
                console.log(error);
            })
    }

    payList() {
        return this.state.pays.map(currentpay => {
            return <Pay pay={currentpay}
                key={currentpay._id}/>;
        })
    }

    render() {
        return (
            <div>
                <h3>Історія сплат</h3>
            </div>
        )
    }
}

```

```

    <table className="table">
      <thead className="thead-light">
        <tr>
          <th>Ім'я</th>
          <th>Прізвище</th>
          <th>Сума, грн</th>
        </tr>
      </thead>
      <tbody>
        {this.payList()}
      </tbody>
    </table>
  </div>
)
}
}

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

const Process = props => (
  <tr>
    <td>{props.process.processtitle}</td>
    <td>{props.process.processdescription}</td>
    <td><a href="#" onClick={() => {
      props.incrementProcessplan(props.process._id)
    }}>
      <button type="submit" className="btn btn-primary" data-toggle="button" aria-pressed="false">
        +
      </button>
    </a>{props.process.processplan} </td>
    <td><a href="#" onClick={() => {
      props.incrementProcessWorkers(props.process._id)
    }}>
      <button type="submit" className="btn btn-primary" data-toggle="button" aria-pressed="false">
        +
      </button>
    </a>{props.process.processWorkers} </td>
    <td>
      <a href="#" onClick={() => {
        props.deleteProcess(props.process._id)
      }}>
        <button type="submit" className="btn btn-primary" data-toggle="button" aria-pressed="false">
          Завершити
        </button>
      </a>
    </td>
  </tr>
)

export default class ProcessList extends Component {
  constructor(props) {
    super(props);

    this.deleteProcess = this.deleteProcess.bind(this)
    this.incrementProcessYes = this.incrementProcessYes.bind(this)
    this.incrementProcessWorkers = this.incrementProcessWorkers.bind(this)
  }
}

```

```

this.onChangeProcesstitle = this.onChangeProcesstitle.bind(this);
this.onChangeProcessdescription = this.onChangeProcessdescription.bind(this);
this.onChangeProcessplan = this.onChangeProcessplan.bind(this);
this.onChangeProcessWorkers = this.onChangeProcessWorkers.bind(this);
this.onSubmit = this.onSubmit.bind(this)

this.state = {
  processtitle: "",
  processdescription: "",
  processplan: '0',
  processWorkers: '0',
  process: [],
};
}

componentDidMount() {
  axios.get('http://localhost:5000/process/')
    .then(response => {
      this.setState({ process: response.data })
    })
    .catch((error) => {
      console.log(error);
    })
}

deleteProcess(id) {
  axios.delete('http://localhost:5000/process/' + id)
    .then(response => {
      console.log(response.data)
    });

  this.setState({
    process: this.state.process.filter(el => el._id !== id)
  })
}

incrementProcessYes(id) {
  const process = {
    processtitle: this.state.processtitle,
    processdescription: this.state.processdescription,
    processplan: this.state.processplan + 1,
    processWorkers: this.state.processWorkers
  }

  console.log(process);

  axios.post('http://localhost:5000/process/update/' + id)
    .then(res => console.log(res.data));
  this.setState({
    process: this.state.process.filter(el => el._id !== id)
  })

  window.location = '/';
}

incrementProcessWorkers(id) {

```

```

    axios.delete('http://localhost:5000/process/' + id)
      .then(response => {
        console.log(response.data)
      });

    this.setState({
      process: this.state.process.filter(el => el._id !== id)
    })
  }

  onChangeProcesstitle(e) {
    this.setState({
      processtitle: e.target.value
    })
  }

  onChangeProcessdescription(e) {
    this.setState({
      processdescription: e.target.value
    })
  }

  onChangeProcessplan(e) {
    this.setState({
      processplan: e.target.value
    })
  }

  onChangeProcessWorkers(e) {
    this.setState({
      processWorkers: e.target.value
    })
  }

  onSubmit(e) {
    e.preventDefault();

    const process = {
      processtitle: this.state.processtitle,
      processdescription: this.state.processdescription,
      processplan: this.state.processplan,
      processWorkers: this.state.processWorkers,
    }

    console.log(process);

    axios.post('http://localhost:5000/process/add', process)
      .then(res => console.log(res.data));

    window.location = '/process';
  }

  processList() {
    return this.state.process.map(currentprocess => {
      return <Process process={currentprocess} deleteProcess={this.deleteProcess}
        key={currentprocess._id}/>;
    })
  }

```

```
}
```

```
render() {  
  return (  
    <div>  
      <div>  
        <h3>Робочий процес</h3>  
        <form onSubmit={this.onSubmit}>  
          <div className="form-row">  
            <div className="form-group col-md-3">  
              <label>Процес</label>  
              <input type="text"  
                required  
                value={this.state.processtitle}  
                onChange={this.onChangeProcesstitle}  
                className="form-control"/>  
            </div>  
            <div className="form-group col-md-3">  
              <label>Часова трудомісткість</label>  
              <input type="text"  
                required  
                value={this.state.processdescription}  
                onChange={this.onChangeProcessdescription}  
                className="form-control"/>  
            </div>  
            <div className="form-group col-md-3">  
              <label>Плановий коефіцієнт норми(%)</label>  
              <input type="text"  
                required  
                value={this.state.processplan}  
                onChange={this.onChangeProcessplan}  
                className="form-control"/>  
            </div>  
            <div>  
              <button type="submit" className="btn btn-primary" data-toggle="button"  
                aria-pressed="false"  
                style={{margin: 30}}>  
                Додати  
              </button>  
            </div>  
          </div>  
        </form>  
      </div>  
      <div>  
        <h3>Робочий процес</h3>  
        <table className="table">  
          <thead className="thead-light">  
            <tr>  
              <th>Процес</th>  
              <th>Час на виконання(год)</th>  
              <th>Плановий коефіцієнт норми(%)</th>  
              <th>Кількість працівників</th>  
            </tr>  
          </thead>  
          <tbody>  
            {this.processList()}  
          </tbody>  
        </table>  
      </div>  
    </div>  
  );  
}
```

```

        </tbody>
      </table>
    </div>
  </div>
)
}
}
}

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

const Worker = props => (
  <tr>
    <td>{props.worker.workername}</td>
    <td>{props.worker.workerlastname}</td>
    <td>{props.worker.workerprof}</td>
    <td>{props.worker.workereffect}</td>
    <td>
      <a href="#" onClick={() => {
        props.deleteWorker(props.worker._id)
      }}>
        <button type="submit" className="btn btn-primary" data-toggle="button" aria-
pressed="false">
          Звільнити робітника {props.worker.workerlastname}
        </button>
      </a>
    </td>
    <td>
      <a href="#" onClick={() => {
        props.makeDuty(props.worker._id)
      }}>
        <button type="submit" className="btn btn-primary" data-toggle="button" aria-pressed="false"
style={{marginRight: 20}}>
          На додаткову зміну
        </button>
      </a>
    </td>
  </tr>
)

export default class WorkerList extends Component {
  constructor(props) {
    super(props);

    this.deleteWorker = this.deleteWorker.bind(this)
    this.onChangeWorkername = this.onChangeWorkername.bind(this);
    this.onChangeWorkerlastname = this.onChangeWorkerlastname.bind(this);
    this.onChangeWorkerprof = this.onChangeWorkerprof.bind(this);
    this.onChangeWorkereffect = this.onChangeWorkereffect.bind(this);
    this.onSubmit = this.onSubmit.bind(this)
    this.makeDuty = this.makeDuty.bind(this)

    this.state = {
      workername: "",
      workerlastname: "",
      workerprof: "",

```

```

        workereffect: "",
        workers: [],
    };
}

componentDidMount() {
    axios.get('http://localhost:5000/worker/')
        .then(response => {
            this.setState({ workers: response.data })
        })
        .catch((error) => {
            console.log(error);
        })
}

deleteWorker(id) {
    axios.delete('http://localhost:5000/worker/' + id)
        .then(response => {
            console.log(response.data)
        });

    this.setState({
        workers: this.state.workers.filter(el => el._id !== id)
    })
}

onChangeWorkername(e) {
    this.setState({
        workername: e.target.value
    })
}

onChangeWorkerlastname(e) {
    this.setState({
        workerlastname: e.target.value
    })
}

onChangeWorkerprof(e) {
    this.setState({
        workerprof: e.target.value
    })
}

onChangeWorkereffect(e) {
    this.setState({
        workereffect: e.target.value
    })
}

onSubmit(e) {
    e.preventDefault();

    const worker = {
        workername: this.state.workername,
        workerlastname: this.state.workerlastname,
    }
}

```



```

</div>
<div className="form-effect col-md-3">
  <label>Прізвище</label>
  <input type="text"
    required
    value={ this.state.workerlastname }
    onChange={ this.onChangeWorkerlastname }
    className="form-control"/>
</div>
<div className="form-effect col-md-3">
  <label>Спеціальність</label>
  <input type="text"
    required
    value={ this.state.workerprof }
    onChange={ this.onChangeWorkerprof }
    className="form-control"/>
</div>
<div className="form-effect col-md-3">
  <label>Часова ефективність</label>
  <input type="text"
    required
    value={ this.state.workereffect }
    onChange={ this.onChangeWorkereffect }
    className="form-control"/>
</div>
<div>
  <button type="submit" className="btn btn-primary" data-toggle="button"
    aria-pressed="false"
    style={{ margin: 30 }}>
    Додати
  </button>
</div>
</div>
</form>
</div>
<div>
  <h3>Працівники</h3>
  <table className="table">
    <thead className="thead-light">
      <tr>
        <th>Ім'я</th>
        <th>Прізвище</th>
        <th>Спеціальність</th>
        <th>Часова ефективність</th>
      </tr>
    </thead>
    <tbody>{ this.workerList()}</tbody></table> </div> </div> ) } }

```

ДОДАТОК Б  
Демонстраційні матеріали

