



Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Звіт до комп'ютерного практикуму №5

З дисципліни «Основи Back-end технологій»

Прийняв:

Зубко Роман Анатолійович

Виконав:

Студент 3 курсу,
Котенко Ярослав
гр. ІМ-13

2023 р.

Лабораторна робота №5. NodeJS.

Робота з БД MongoDB. Додаток що реалізує CRUD операції в БД.

Завдання.

- Створити додаток, що реалізує CRUD операції з БД – додавання, читання, редагування та видалення записів БД.
- Забезпечити роутінг запитів та виведення результатів запитів на WEBсторінку.
- Додати новий роут для виведення інформації у вигляді json-файлу.

Варіант 11. Спроектувати базу даних технологічних карт: деталь, вид обробки, тривалість обробки.

Хід роботи

Розробив додаток який реалізує CRUD операції з базою даних. Забезпечив роутінг запитів та виведення результатів на WEB сторінку.

Код програми

Файл: *index.js*

```
JS index.js  ×
JS index.js > ...
1  import express from 'express'
2  import mongoose from 'mongoose'
3  import path from 'path'
4  import { fileURLToPath } from 'url'
5  import { create } from 'express-handlebars'
6  import Handlebars from 'handlebars'
7  import { allowInsecurePrototypeAccess } from '@handlebars/allow-prototype-access'
8
```

```

9   import mainRoute from './routes/main.js'
10  import addCardRoute from './routes/addCard.js'
11  import showCardsRoute from './routes/cards.js'
12  import editCardRoute from './routes/editCard.js'
13  import deleteCardRoute from './routes/deleteCard.js'
14  import jsonCardRoute from './routes/cardJson.js'
15
16  const url = 'mongodb+srv://yarikkotenkoim13:LUGD3nhtc1ZpvEST@cluster0.qbd8tfw.mongodb.net/technological_cards'
17  const PORT = process.env.PORT || 5000
18
19  const app = express()
20
21  const hbs = create({
22    defaultLayout: 'main',
23    extname: 'hbs',
24    handlebars : allowInsecurePrototypeAccess(Handlebars)
25  })
26
27  const __filename = fileURLToPath(import.meta.url)
28  const __dirname = path.dirname(__filename)
29
30  app.engine('hbs', hbs.engine)
31  app.set('view engine', 'hbs')
32  app.set('views', 'views')
33  app.use(express.static(path.resolve(__dirname, 'public')))
34  app.use(express.urlencoded({extended: true}))
35  app.use(express.json())
36
37  app.use('/', mainRoute)
38  app.use('/cards', showCardsRoute)
39  app.use('/add', addCardRoute)
40  app.use('/edit', editCardRoute)
41  app.use('/delete', deleteCardRoute)
42  app.use('/json', jsonCardRoute)
43
44  const start = async () => {
45    await mongoose.connect(url)
46    app.listen(PORT, () => {
47      console.log(`Server is running on port: ${PORT}`)
48    })
49  }
50
51  start()

```

index.js є головним файлом. У ньому відбувається підключення всіх необхідних залежностей для функціонування застосунку, а також налаштовані маршрути.

Так само, як і у попередній лабораторній роботі, я використовував шаблонізатор Express Handlebars для створення веб-інтерфейсу.

Файл: technologicalCard.js

```
JS technologicalCard.js x
models > JS technologicalCard.js > technologicalCardSchema > detailName
1 import { model, Schema } from 'mongoose'
2
3 const technologicalCardSchema = new Schema({
4   detailName: {
5     type: String,
6     required: true
7   },
8   processTypeName: {
9     type: String,
10    required: true
11  },
12  duration: {
13    type: Number,
14    required: true
15  },
16  createDate: {
17    type: Date,
18    default: Date.now
19  },
20  description: {
21    type: String,
22    required: true
23  }
24 })
25
26 export default model('technologicalCard', technologicalCardSchema)
```

Створив модель у Mongoose для колекції "Technological cards".

Файл: main.js

```
JS main.js x
routes > JS main.js > ...
1 import { Router } from 'express'
2 import Controller from './controller/Controller.js'
3
4 const router = Router()
5
6 router.get('/', Controller.getMainPage)
7
8 export default router
```

Налаштував маршрут для головної сторінки (route) в моєму додатку.

Файл: *cards.js*

```
JS cards.js ×
routes > JS cards.js > [?] default
1  import { Router } from 'express'
2  import Controller from './controller/Controller.js'
3
4  const router = Router()
5
6  router.get('/', Controller.getCardsPage)
7  router.get('/:id', Controller.getCardById)
8
9  export default router
```

Маршрут для сторінки з картками.

Файл: *addCard.js*

```
JS addCard.js ×
routes > JS addCard.js > [?] default
1  import { Router } from 'express'
2  import Controller from './controller/Controller.js'
3
4  const router = Router()
5
6  router.get('/', Controller.getAddCardPage)
7  router.post('/', Controller.createNewCard)
8
9  export default router
```

Маршрут для створення нової картки.

Файл: *deleteCard.js.js*

```
JS deleteCard.js ×
routes > JS deleteCard.js > ...
1  import { Router } from 'express'
2  import Controller from './controller/Controller.js'
3
4  const router = Router()
5
6  router.get('/:id', Controller.getDeleteCardPage)
7  router.delete('/:id', Controller.deleteCard)
8
9  export default router
```

Маршрут для видалення картки.

Файл: *editCard.js.js*

```
JS editCard.js X
routes > JS editCard.js > [🔍] default
1  import { Router } from 'express'
2  import Controller from './controller/Controller.js'
3
4  const router = Router()
5
6  router.get('/:id', Controller.getEditCardPage)
7  router.put('/:id', Controller.editCard)
8
9
10 export default router
```

Маршрут для редагування картки

Файл: *cardJson.js.js*

```
JS cardJson.js X
routes > JS cardJson.js > [🔍] default
1  import { Router } from 'express'
2  import Controller from './controller/Controller.js'
3
4  const router = Router()
5
6  router.get('/', Controller.getAllInfoJSON)
7  router.get('/:id', Controller.getCardInfoJSON)
8
9  export default router
```

Маршрут для зображення карток у форматі json.

Файл: *Controller.js*

```
JS Controller.js ×
routes > controller > JS Controller.js > Controller > editCard
1  import technologicalCard from "../../models/technologicalCard.js"
2
3  export default class Controller {
4    static getMainPage(req, res) {
5      res.render('index', {
6        title: 'Home page',
7        isMain: true
8      })
9    }
10   static getAddCardPage(req, res) {
11     res.render('addCard', {
12       title: 'Add new card',
13       cardAddedMessage: req.query.cardAdded,
14       isAdd: true
15     })
16   }
17   static async getCardsPage(req, res) {
18     const technologicalCards = await technologicalCard.find()
19     res.render('cards', {
20       title: 'Technological cards',
21       technologicalCards,
22       isCards: true
23     })
24   }
25   static async createNewCard(req, res) {
26     try {
27       const { detailName, processTypeName, duration, creationDate, description } = req.body
28       const newTechCard = new technologicalCard({
29         detailName, processTypeName, duration, creationDate, description
30       })
31       await newTechCard.save()
32       res.redirect('/add?cardAdded=true')
33     } catch (e) {
34       console.log(e)
35     }
36   }
```

```
37   static async getCardById(req, res) {
38     const selectedTechCard = await technologicalCard.findById(req.params.id)
39     res.render('techCard', {
40       title: 'Tech card',
41       selectedTechCard,
42       isTechCard: true
43     })
44   }
45
46   static async getEditCardPage(req, res) {
47     const selectedTechCard = await technologicalCard.findById(req.params.id)
48     let dateToStr = formatDate(selectedTechCard.creationDate)
49     res.render('editCard', {
50       title: 'Edit card',
51       selectedTechCard,
52       dateToStr,
53       isEdit: true
54     })
55
56     function formatDate(date) {
57       const d = new Date(date);
58       const year = d.getFullYear();
59       let month = '' + (d.getMonth() + 1);
60       let day = '' + d.getDate();
61       if (month.length < 2) {
62         month = '0' + month;
63       }
64       if (day.length < 2) {
65         day = '0' + day;
66       }
67       return [year, month, day].join('-');
68     }
69   }
70   static async editCard(req, res) {
```

```

68     }
69   }
70   static async editCard(req, res) {
71     try {
72       const cardId = req.params.id
73       delete req.body.cardId
74       await technologicalCard.findByIdAndUpdate(cardId, req.body)
75       res.json({status: 0, cardId})
76     } catch(e) {
77       res.status(500).json({ status : 1 , error: e })
78     }
79   }
80
81   static async getDeleteCardPage(req, res) {
82     res.render('deleteCard', {
83       title: 'Delete card',
84       layout: 'deleteConfirm',
85       cardId: req.params.id,
86       isDelete: true
87     })
88   }
89
90   static async deleteCard(req, res) {
91     try {
92       const cardId = req.params.id
93       await technologicalCard.findByIdAndDelete(cardId)
94       res.json({ status : 0 })
95     } catch(e) {
96       res.status(500).json({ status : 1 , error: e })
97     }
98   }
99
100  static async getCardInfoJSON(req,res) {
101    const cardId = req.params.id
102    const selectedCard = await technologicalCard.findById(cardId)
103    res.json(selectedCard)
104  }
105
106  static async getAllInfoJSON(req,res) {
107    const cards = await technologicalCard.find()
108    res.json(cards)
109  }
110 }

```

Цей файл містить всі функції, які використовуються у маршрутах цього додатку.

Файл: *delete.js*

```

JS delete.js x
public > JS delete.js > confirmButton.addEventListener('click') callback > then() callback
1  const hiddenInput = document.querySelector('input[type="hidden"][name="id"]')
2  const confirmButton = document.querySelector('input[type="button"][value="Yes"]');
3
4  const cardId = hiddenInput.value
5
6  confirmButton.addEventListener('click', (event) => {
7    fetch(`/delete/${cardId}`, {
8      method: 'DELETE',
9    })
10   .then(res => res.json())
11   .then(res => {
12     if(res.status === 0) {
13       window.location.href = '/cards'
14     } else {
15       console.log(res.error)
16     }
17   })
18 })

```


Клієнтський скрипт, який надсилає запит DELETE на сервер.

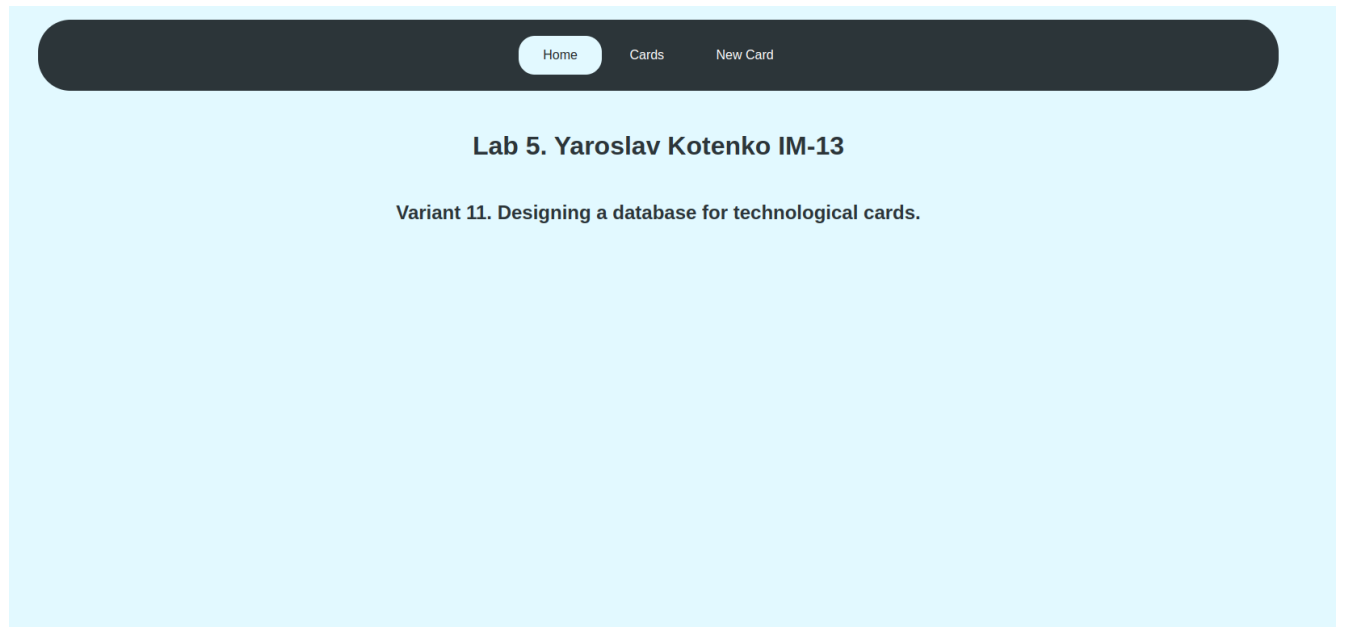
Файл: *edit.js*

```
JS edit.js
public > JS edit.js > ...
1  const submitButton = document.querySelector('button[type="button"]')
2
3  submitButton.addEventListener('click', (event) => {
4    const form = document.querySelector('.edit-form');
5
6    const cardId = form.querySelector('#cardId').value;
7    const detailName = form.querySelector('#detailName').value;
8    const processTypeName = form.querySelector('#processTypeName').value;
9    const duration = form.querySelector('#duration').value;
10   const creationDate = form.querySelector('#creationDate').value;
11   const description = form.querySelector('#description').value;
12
13   const updatedData = {
14     detailName,
15     processTypeName,
16     duration,
17     creationDate,
18     description
19   };
20
21   fetch(`/edit/${cardId}`, {
22     method: 'PUT',
23     headers: {
24       'Content-Type': 'application/json'
25     },
26     body: JSON.stringify(updatedData)
27   })
28   .then(res => res.json())
29   .then(res => {
30     if(res.status === 0) {
31       window.location.href = `/cards/${cardId}`
32     } else {
33       console.log(res.error)
34     }
35   })
36 })
37
```

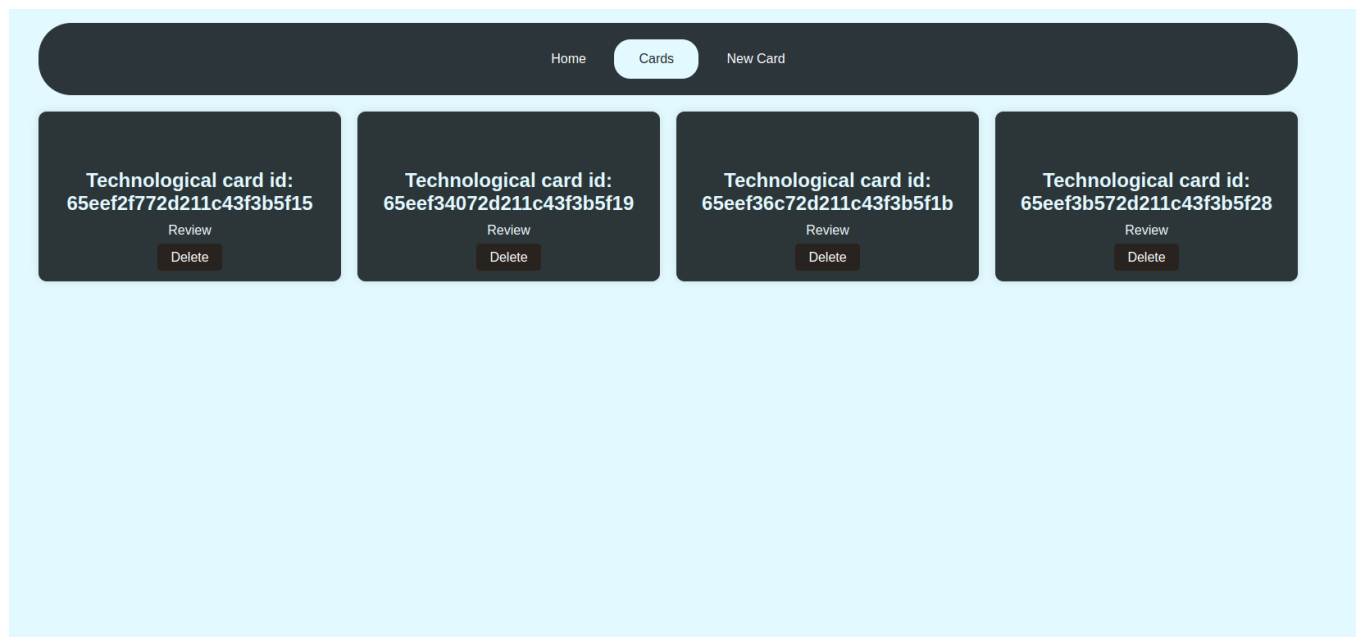
Клієнтський скрипт, який надсилає запит PUT на сервер.

Скріншоти виконання коду

Головна сторінка



Сторінка з усіма технологічними картками



Сторінка створення нової технологічної картки

[Home](#) [Cards](#) [New Card](#)

Detail name:

Process Type name:

Duration:

Creation Date:

Description:

Submit

Сторінка перегляду технологічної картки

[Home](#) [Cards](#) [New Card](#)

Object Technological card

card_ID: 65eef34072d211c43f3b5f19

Detail Name: M8 Bolt

Process Type Name: Turning

Duration: 45

Creation Date: Sat Mar 09 2024 02:00:00 GMT+0200 (Eastern European Standard Time)

Description: Description of the technological card for the bolt and turning processing

Edit

Сторінка редагування технологічної картки

[Home](#) [Cards](#) [New Card](#)

Edit tech card with id: 65eef34072d211c43f3b5f19

Detail name:

M8 Bolt

Process Type name:

Turning

Duration:

45

Creation Date:

2024-03-09

Description:

Description of the technological card for the bolt and turning processing

Submit

Back

Сторінка видалення технологічної картки

Do you really want to remove the technology card with id: 65eef34072d211c43f3b5f19?

Yes

No

Висновок:

Під час виконання лабораторної роботи я навчився працювати з базою даних MongoDB, виконувати операції CRUD за допомогою Mongoose. Покращив свої навички у розробці серверної частини за допомогою Express, обробці HTTP-запитів та налаштуванні роутингів. Крім того, покращив навички роботи з клієнтською частиною, створюючи інтерфейси для взаємодії з користувачем та відправляючи AJAX-запити на сервер. Виконання цієї роботи

дало мені досвід у розробці повноцінних веб-додатків і покращило мої навички веб-розробника на Node.js.