Лабораторна робота №5.

Зв'язування користувачів і задач

Мета: навчитись зв'язувати дані. Завершення виконання TaskApp

Завдання

- Створити відношення між користувачами та задачами
- Забезпечити роботу користувача лише з власними задачами (GET /tasks, GET /tasks/:id, ...)
- Приховати захищені дані (pasword, tokens)

Завдання 1. Зв'язок між користувачами та задачами

Реалізуйте можливість доступу та обробки лише задач поточного користувача

Крок 1. В моделі Task створіть поле для зберігання іd користувача

```
default: false
},

owner: {
    type: mongoose.Schema.Types.ObjectId,
    ref: "User",
    required: true
}
```

					ПV «Житомипська політех:	ДУ «Житомирська політехніка».24.121.13.000 - Лр5			
Змн.	Арк.	№ докум.	Підпис	Дата	д «житомирськи політехніки».24.121.13.000 - Прэ				
Розра	об.	Клосович І.А.			Звіт з	Літ.	Арк.	Аркушів	
Пере	вір.	Сидорчук В.О.			3611.3		1	18	
Керів	вник				лабораторної роботи				
Н. контр.						ФІКТ Гр. ІПЗ-22-1[1]			
Зав. 1	каф.						•		

```
const Task = new mongoose.Schema({

✓ Lab3_4_5_TaskApp

                                                                title: {
∨ db
 JS mongoose.js
                                                                description: { --

✓ models

                                                                completed: { ···
 JS task.js
 JS user.js
                                                                owner: {
                                                                    type: mongoose.Schema.Types.ObjectId,
∨ src
                                                                    ref: 'User',

✓ middleware

                                                                    required: true
  JS auth.js
                                                     23

✓ routers
```

Рис. 1. Результат

Крок 2. В маршрутизаторі підключіть auth та модифікуйте обробник додавання нового запису

```
POST
                                                                                     ▼ {{url}}/tasks
prouter.post("/tasks", auth, async (req, res) => {
                                                                           Params
                                                                                  Authorization Headers (10) Body ●
        //let task = new Task(req.body);
        const task = new Task({
                                                                            none form-data x-www-form-urlencoded raw
              ...req.body,
              owner: req.user.id
                                                                          4 }
        });
        try {
              await task.save();
             res.status(201).send(task);
                                                                          Body Cookies Headers (6) Test Results
        } catch (e) {
              \underline{\text{res}}.status(500).send(\underline{e});
                                                                           "completed": true,
"_id": "5cd2975ff1194728dcc41774",
"description": "New Task",
"owner": "5cd29752f1194728dcc41772",
 });
                                                                           7 }
```

```
✓ Lab3_4_5_TaskA...

✓ db

                               router.post('/tasks', auth, async (req, res) => {
                                   try {
 JS mongoose.js
                                        const task = new Task({...req.body, owner: {

✓ models

                                            _id: req.user._id
 JS task.js
                         32
                                        }});
 JS user.js
                                        await task.save();
> node_modules
                                        res.status(201).send(task);
∨ src
                                    } catch (error) {
                                        res.status(500).send(error);

✓ middleware

  JS auth.js
                               });

✓ routers
```

Рис. 2. Результат

		Клосович І.А.		
		Сидорчук В.О.		
Змн.	Арк.	№ докум.	Підпис	Дата

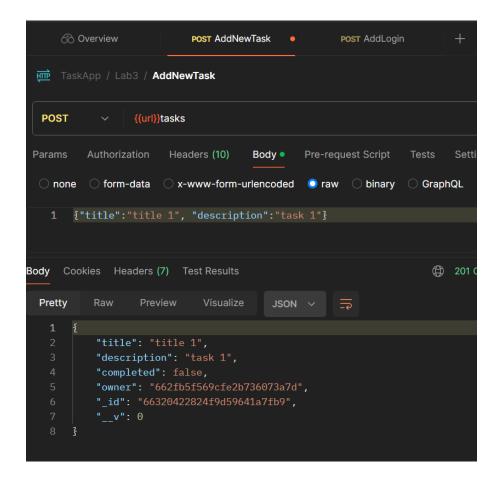


Рис. 3. Результат

Крок 3. Для отримання запису користувача, якому належить дана задача:

• Приклад коду:

```
const task = await Task.findById('5cd2975ff1194728dcc41774');
await task.populate('owner').execPopulate();
```

В нових версіях mongoose: await task.populate('owner');

		Клосович І.А.		
		Сидорчук В.О.		
Змн.	Арк.	№ докум.	Підпис	Дата

```
∨ Lab3_4_5_TaskA... ●
                               router.get('/tasks/:id', auth, async (req, res) => {

✓ db

                                    try {
 JS mongoose.js
                                        const task = await Task.findById(req.params.id)

✓ models

                                        await task.populate('owner')
                         21
 JS task.js
                 М
                                        res.status(200).send(task)
 JS user.js
                                    catch (error) {
                                        res.status(500).send(error)

✓ src

✓ middleware

                               });
  JS auth.js
```

Рис. 4. Результат

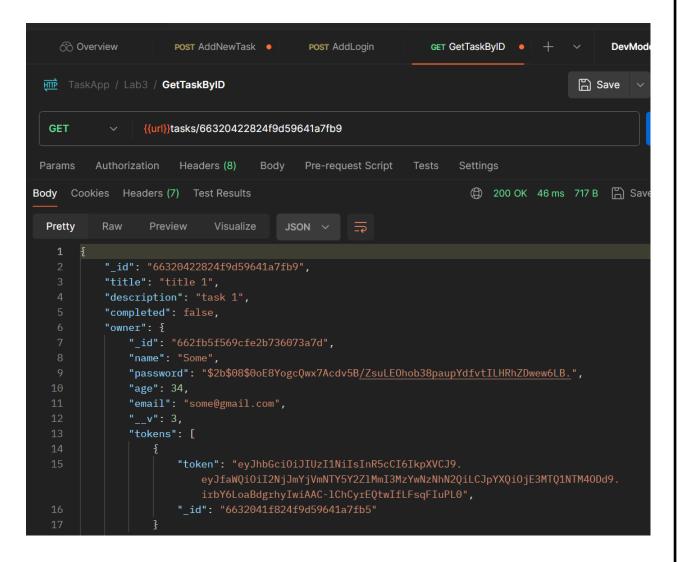


Рис. 5. Результат

		Клосович І.А.			
		Сидорчук В.О.			ДУ «Житомирська політехніка».24.121.13.000 - Лр5
Змн.	Арк.	№ докум.	Підпис	Дата	

Крок 4. Для отримання задач, які належать даному користувачу, створіть віртуальне поле в моделі User. Відобразіть дані поля tasks в профілі користувача (запит /users/me)

```
userSchema.virtual('tasks', {
    ref: "Task",
    localField: '_id',
    foreignField: 'owner'
})
```

const userSchema = new mongoose.Schema({
 firstName: {type: String...},
 lastName: {type: String...},
 age: {type: Number...},
 email: {type: String...},
 password: {type: String...},
 tokens: [{...}]
}, {toJSON: {virtuals: true}})

• Приклад коду для виконання:

```
const user = await User.findById('5cd29752f1194728dcc41772');
await user.populate("tasks").execPopulate();
```

В нових версіях mongoose: await user.populate('tasks');

```
Lab3_4_5_laskApp / models / 🍱 user.js / 🖾 userSchema
                              const userSchema = new mongoose.Schema({
Lab1
                                  age: { ···
Lab2_WeatherApp
Lab3 4 5 TaskA...
                                  email: { ···
db '
                                  tokens: [{ ···
JS mongoose.js
✓ models
                             }, { toJSON: { virtuals: true } })
                       53
JS task.js
               M
               М
JS user.js
                             userSchema.virtual('tasks', {
                                  ref: 'Task',
                                  localField: ' id',
                       58
                                  foreignField: 'owner'

✓ middleware

JS auth.js

✓ routers

                     JS user.js M X
JS task.js
                      Lab3_4_5_TaskApp > src > routers > JS user.js > ♥ router.get('/users/:id') callback
 JS user.js
                             router.get('/users', auth, async (req, res) => {
JS app.js
                             });
 .env
                             router.get('/users/:id', auth, async (req, res) => {
 eslint.config.mjs
package-lock.json
                                      const user = await User.findById(req.params.id)
package.json
                                      await user.populate('tasks')
                       26
reports
                                      res.status(200).send(user)
.gitignore
                                  catch (error){
                                      res.status(500).send(error)
                                  OUTPUT DEBUG CONSOLE
                                                           TERMINAL
```

Рис. 6. Результат

		Клосович I.A.		
		Сидорчук В.О.		
Змн.	Арк.	№ докум.	Підпис	Дата

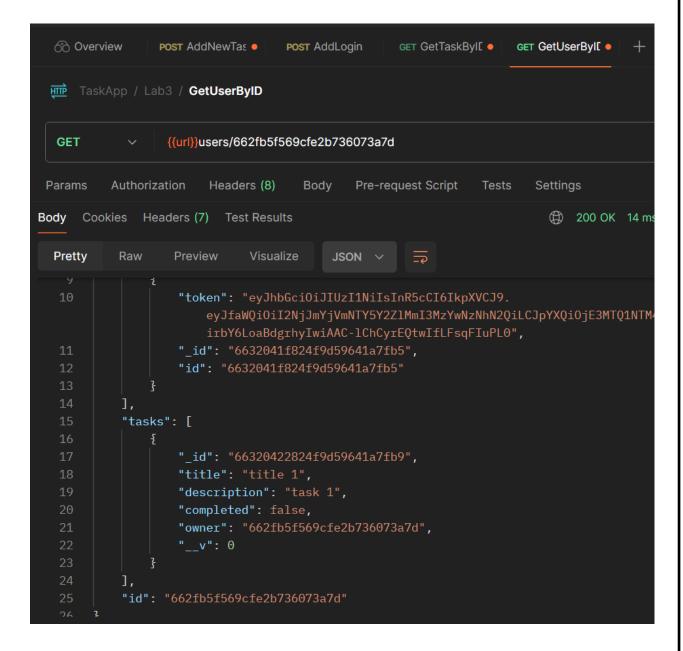


Рис. 7. Результат

Завдання 2. Реалізувати можливість роботи користувачам лише з власними завданнями

- Реалізуйте отримання задачі по її іd лише для користувача, якому належить дана задача. Для авторизованого користувача, якому не належить дана задача, надіслати помилку 404.
- Модифікуйте метод отримання списку задач, лише таких, що належать даному користувачу
- Створіть методи редагування/видалення задачі лише тої, що належить даному користувачу. Надіслати 404, якщо задачі не існує або вона не належить поточному користувачеві

		Клосович I.A.		
		Сидорчук В.О.		
Змн.	Арк.	№ докум.	Підпис	Дата

```
> Lab2_WeatherApp
                              router.get('/tasks', auth, async (req, res) => {
∨ Lab3_4_5_TaskA... ●
 ∨ db
                                       const tasks = await Task.find({ owner: {
                                           _id: req.user._id
 JS mongoose.js

✓ models

                                       res.send(tasks);
 JS task.js
                                  } catch (error) {
               М
 JS user.js
                                      res.status(500).send(error);
 ∨ src

✓ middleware

                              router.get('/tasks/:id', auth, async (req, res) => {
  JS auth.js
 routers
                                       const task = await Task.findById(req.params.id)
  JS task.js
                        21
                                       if (!task) {
  JS user.js
               M
                                           throw new Error("Not existing")
 JS app.js
.env
                                       const belongs = task.owner._id.toString() === req.user._id.toString()
                                       if (!belongs) {
JS eslint.config.mjs
                                           return res.status(404).send({
{} package-lock.json
                                               error: "IdError",
message: "This task is not yours"
{} package.json
> reports
gitignore
                        31
                                       await task.populate('owner')
                                       res.status(200).send(task)
                                   catch (error) {
                                       res.status(500).send(error)
```

Рис. 8. Результат

```
> Lab2_WeatherApp

✓ Lab3_4_5_TaskA...

                              router.delete('/tasks/:id', auth, async (req, res) => {
                                  const taskId = req.params.id;
 ∨ db
 JS mongoose.js
                                       const task = await Task.findById(taskId);

✓ models

                                       if (task.owner._id.toString() !== req.user._id.toString()) {
 JS task.js
                                           return res.status(404).send({
 JS user.js
                M
                                               error: "IdError",
                                               message: "This task is not yours"
 ∨ src

✓ middleware

                                       if (!task) {
  JS auth.js
                                           return res.status(404).send("Task not found");

✓ routers

  JS task.js
                                       await Task.deleteOne(task)
  JS user.js
                                       res.status(200).send(task);
 JS app.js
                                   } catch (error) {
                                       console.error("Error deleting task:", error);
.env
                                       res.status(500).send("Internal Server Error");
JS eslint.config.mjs
{} package-lock.json
{} package.json
```

Рис. 9. Результат

		Клосович I.A.			
		Сидорчук В.О.			ДУ «Житомирська політехніка».24.121.13.000 - Лр5
Змн.	Арк.	№ докум.	Підпис	Дата	

```
res.status(500).send("Internal Server
 Lab2_WeatherApp
∨ Lab3_4_5_TaskA... ●

✓ db

 JS mongoose.js
                              router.patch('/tasks/:id', auth, async (req, res, next) => {

✓ models

                       84
                                  const taskId = req.params.id;
 JS task.js
                                  try {
                                      const task = await Task.findById(taskId);
 JS user.js
                                      if (task.owner._id.toString() !== req.user._id.toString()) {
                                          return res.status(404).send({
∨ src

✓ middleware

                                              message: "This task is not yours"
  JS auth.js

✓ routers

                                      if (!task) {
  JS task.js
                                           return res.status(404).send("Task not found");
              M
 JS user.js
 JS app.js
                                      const updateData = req.body
.env
                                      await Task.updateOne(task, updateData)
JS eslint.config.mjs
                                      res.status(200).send(task);
{} package-lock.json
                                      next(e)
{} package.json
> reports
```

Рис. 10. Результат

Послідовність тестів

- 1. Реєструємо два користувача user1, user2
- 2. Список задач. Повинна бути помилка 403 (не авторизовано)
- Логін user1.
- 4. Створюємо дві задачі і переглядаємо їх. Записуємо їх ідентифікатори.
- 5. Здійснюємо вихід user1.
- 3аходимо під user2.
- 7. Створюємо дві задачі. Переглядаємо їх, редагуємо, видаляємо.
- Спробуємо переглянути або змінити задачі user1. Повинна бути помилка 404.
- 9. Видаляємо всі задачі і здійснюємо вихід.
- 10. Здійснюємо вхід під user1. Виводимо задачі user1. Вони повинні бути на місці.

		Клосович І.А.		
		Сидорчук В.О.		
Змн.	Арк.	№ докум.	Підпис	Дата

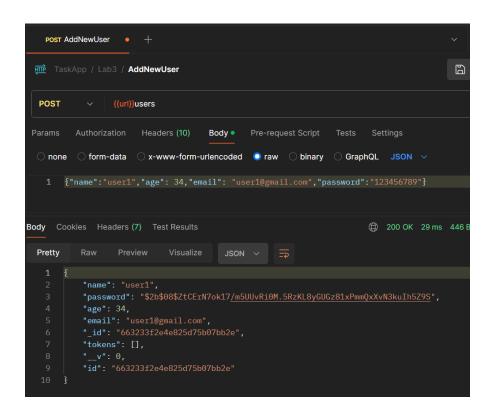


Рис. 11. Результат реєстрації user1(аналогічно створюється user2)

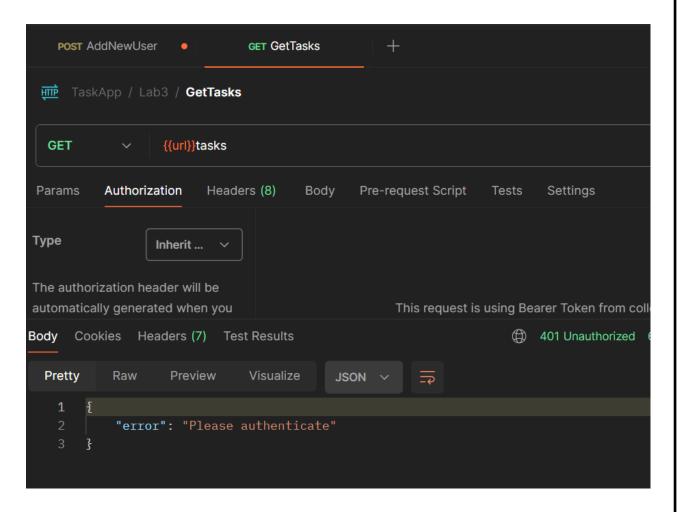


Рис. 12. Список задач

		Клосович I.A.			
		Сидорчук В.О.			ДУ «Житомирська політехніка».24.121.13.000 - Лр5
Змн.	Арк.	№ докум.	Підпис	Дата	

 $Ap\kappa$.

9

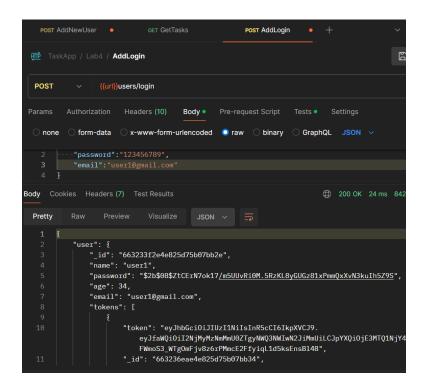


Рис. 13. Логін user1

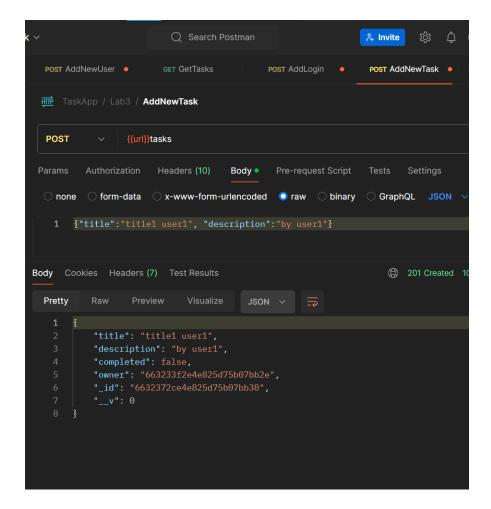


Рис. 14. Створення задачі user1(і ще одної)

		Клосович І.А.			
		Сидорчук В.О.			ДУ «Житомирська політехніка».24.121.13.000 - Лр5
Змн.	Арк.	№ докум.	Підпис	Дата	

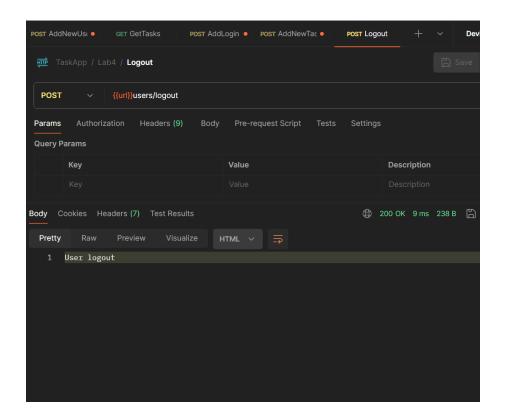


Рис. 15. Buxiд user1

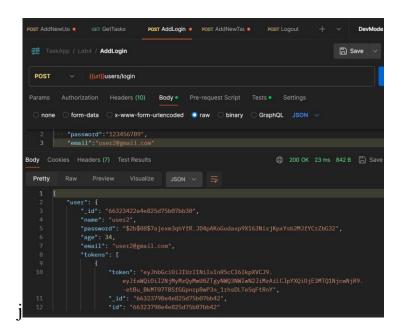


Рис. 16. Bxiд user2

		Клосович І.А.		
		Сидорчук В.О.		
Змн.	Арк.	№ докум.	Підпис	Дата

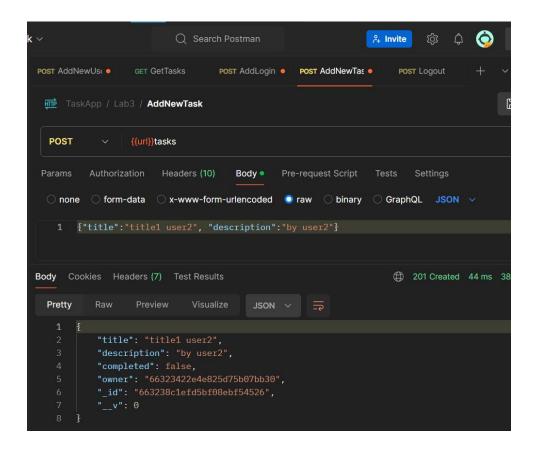


Рис. 17. Створення задачі для user2(+ще одну окремо)

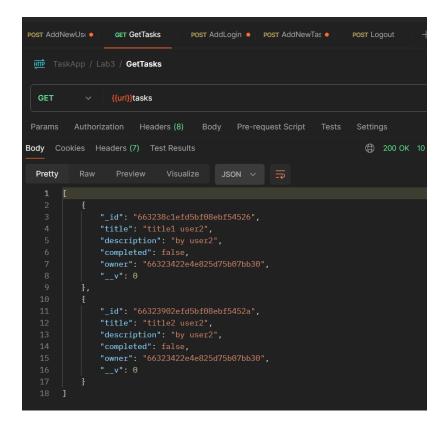


Рис. 18. Перегляд задач тільки user2

		Клосович I.A.		
·		Сидорчук В.О.		
Змн.	Арк.	№ докум.	Підпис	Дата

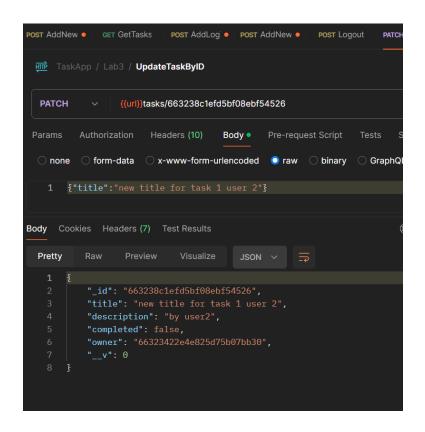


Рис. 19. Редагування першої задачі user2

		Клосович І.А.		
		Сидорчук В.О.		
Змн.	Арк.	№ докум.	Підпис	Дата

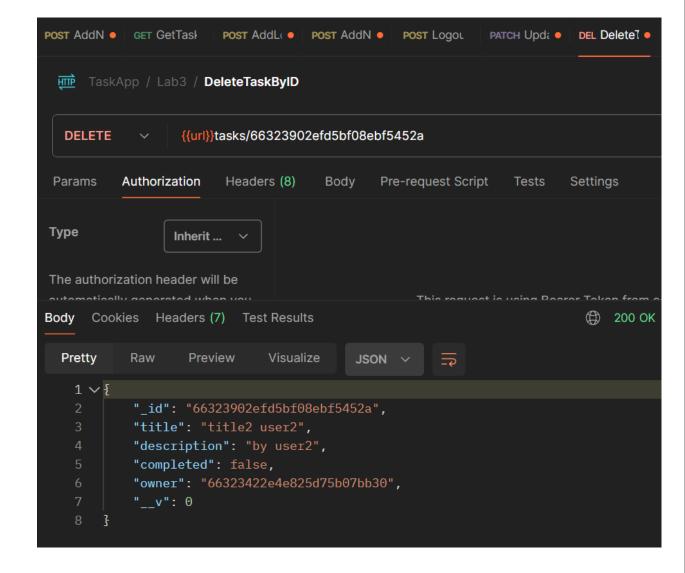


Рис. 20. Результат видалення task2 user2

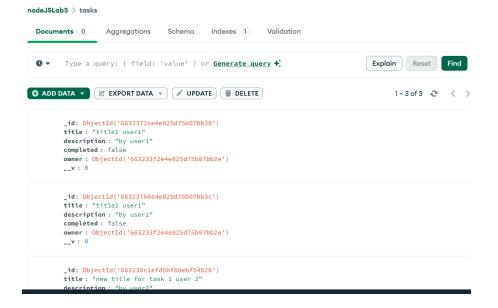


Рис. 21. Результат

		Клосович I.A.			
		Сидорчук В.О.			ДУ «Житомирська політехніка».24.121.13.000 - Лр5
Змн.	Арк.	№ докум.	Підпис	Дата	

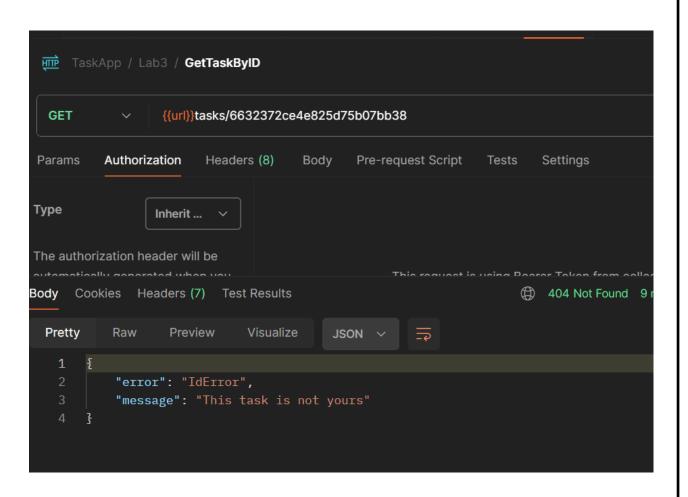


Рис. 22. Перегляд задачі, що належить user1

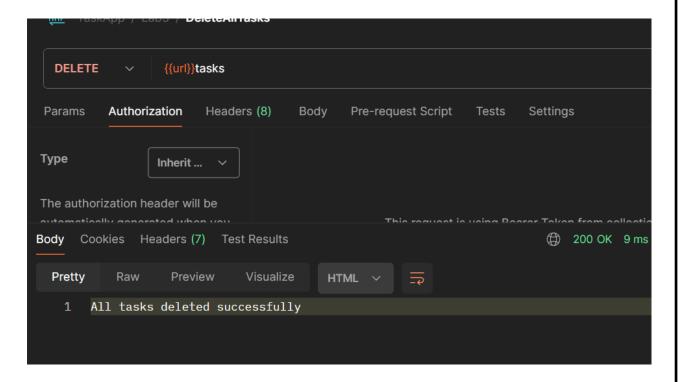


Рис. 23. Видалення всіх задач

		Клосович I.A.			
		Сидорчук В.О.			ДУ «Житоми
Змн.	Арк.	№ докум.	Підпис	Дата	

 $Ap\kappa$.

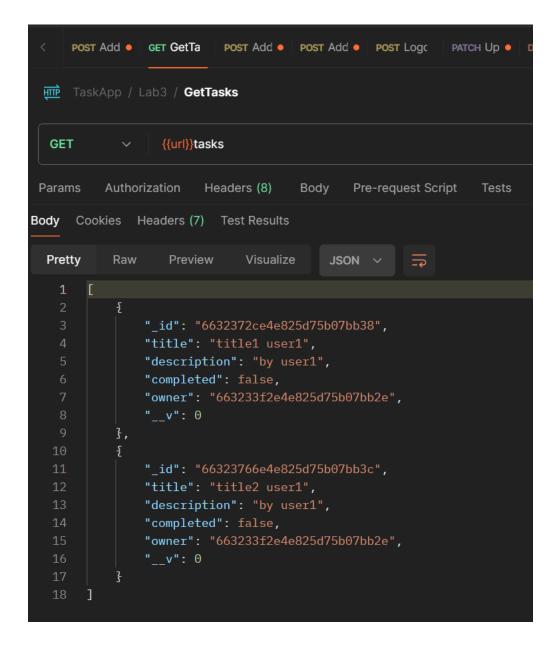


Рис. 24. Вивід задач user1

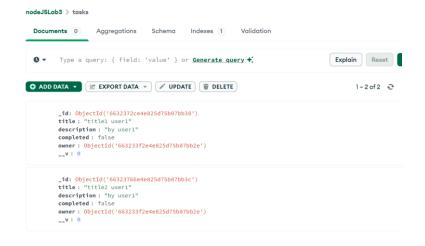


Рис. 25. Результат

		Клосович I.A.			
		Сидорчук В.О.			ДУ «Житомирська політехніка».24.121.13.000 - Лр5
Змн.	Арк.	№ докум.	Підпис	Дата	

 $Ap\kappa$.

16

Завдання 3. Забороніть відправку захищених даних на клієнт:

```
UserSchema.methods.toJSON = function() {
    const user = this;
    const userObject = user.toObject();
    delete userObject.password;
    delete userObject.tokens;
    return userObject;
}
```

Для дозволу передавати віртуальні поля (tasks) в метод toObject, в схемі даних додайте опцію:

```
tokens: [{...}]
}, {toJSON: {virtuals: true}, toObject: {virtuals: true}})
```

```
NODE.JS
                       Lab3_4_5_TaskApp > models > JS user.js > ...
                         7 const userSchema = new mongoose.Schema({
> .idea
> Lab1
                                   tokens: [{ ···
> Lab2_WeatherApp
∨ Lab3_4_5_TaskA... ●
                               }, { toJSON: { virtuals: true }, toObject:{virtuals: true} })
 ∨ db
 JS mongoose.js

✓ models

                       JS user.js M X
 JS task.js
                 М
                       Lab3_4_5_TaskApp > models > JS user.js > ...
 JS user.js
                              userSchema.methods.generateAuthToken = async function () {
 ∨ src

✓ middleware

                         90
                             userSchema.methods.toJSON = function() {
  JS auth.js
                                   const user = this

✓ routers

                                   const userObject = user.toObject()
                        92
                         93
                                   delete userObject.password
  JS task.js
                                   delete userObject.tokens
  JS user.js
                 М
                                   return userObject
 JS app.js
 .env
JS eslint.config.mjs
                               const User = mongoose.model("User", userSchema)
{} package-lock.json
                               module.exports = User
 {} package.json
```

Рис. 26. Результат

		Клосович І.А.			
		Сидорчук В.О.			4
Змн.	Арк.	№ докум.	Підпис	Дата	

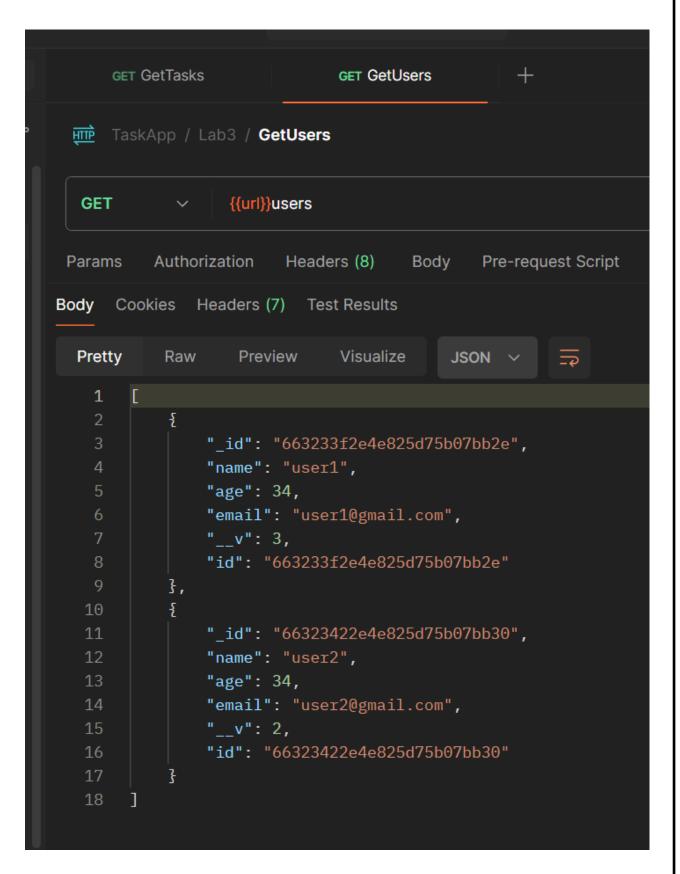


Рис. 27. Результат

Висновок: на лабораторному занятті ми навчились зв'язувати дані.

Завершили виконання TaskApp

		70 7 1			
		Клосович І.А.			
		Сидорчук В.О.			ДУ «Житомирська політехніка».24.121.13.000 - Лр5
Змн.	Арк.	№ докум.	Підпис	Дата	

Арк. 18