Міністерство освіти і науки України
Національний університет „Львівська політехніка”
Кафедра ЕОМ

Лабораторна робота

З дисципліни:”Інженерія Програмного Забезпечення”

«Розробка серверної частини. Розробка комунікації за протоколом TCP. Підключення серверного модуля до БД»

Виконав:
ст.гр. КІ-37
Філіпович Т.І.
Прийняв:
Бойко Г.В.

Львів - 2018

Мета: Розробити консольну аплікацію, що буде підтримувати зв’язок TCP, отримувати дані та записувати в БД. Також, згідно деякої команди, вичитувати з БД необхідну інформацію та передавати по TCP протоколу на клієнтську частину.

Завдання: Розробити консольну аплікацію(серверну частину), яка буде передавати по TCP протоколу, записувати у БД та зчитувати необхідні дані.

Варіант №13: Інтернет-магазин автомобільних крісел

Виконання

Вигляд запущеної програми:

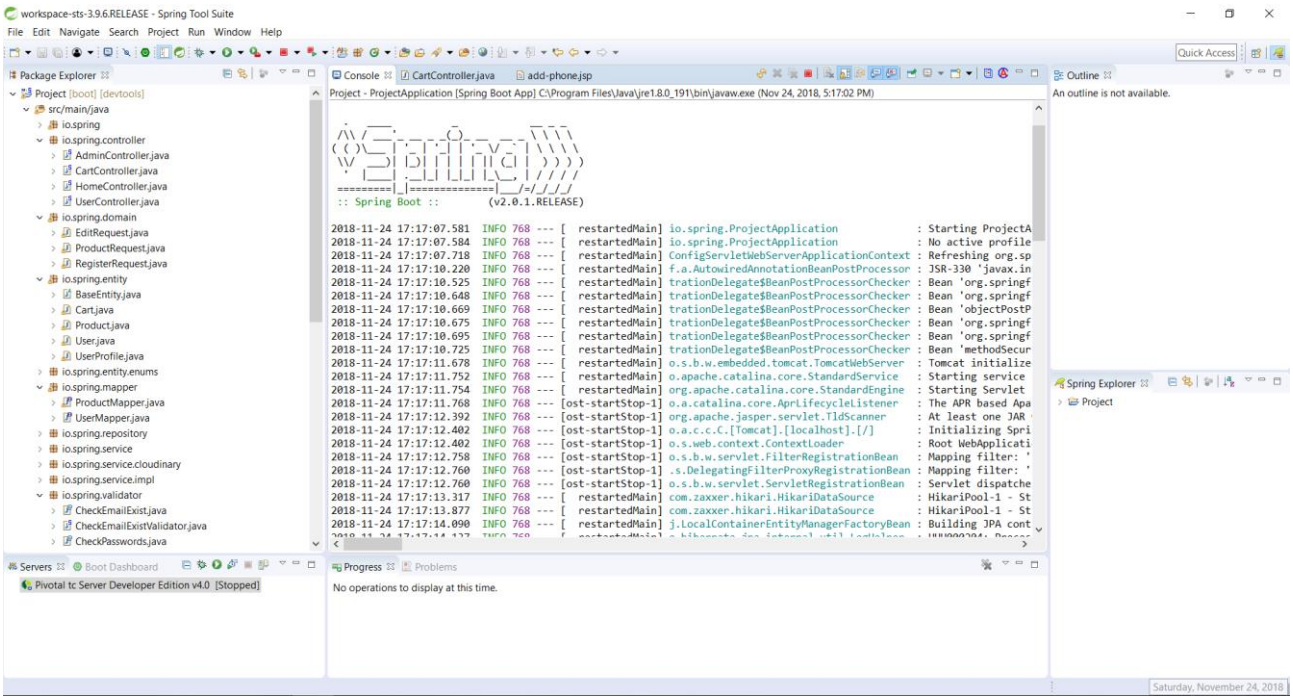

Рис.1. Вікно програми – сервера

Метод **public ModelAndView registerUser(@Valid @ModelAttribute("registerModel") RegisterRequest request, BindingResult br){} -** Виконує реєстрацію користувача і всі дані записуються в БД. Також перевіряє дані на правильність введення.
Метод **public String showAddProduct (Model model){}** – Показує сторінку додавання продукту.
Метод **public ModelAndViewSaveProduct(@ModelAttribute ("productAddModel") Product product)** – зберігає продукт в БД.
Метод **private String showProduct (Model model){}–** Показує сторінку всіх продуктів.

Вигляд бази даних:

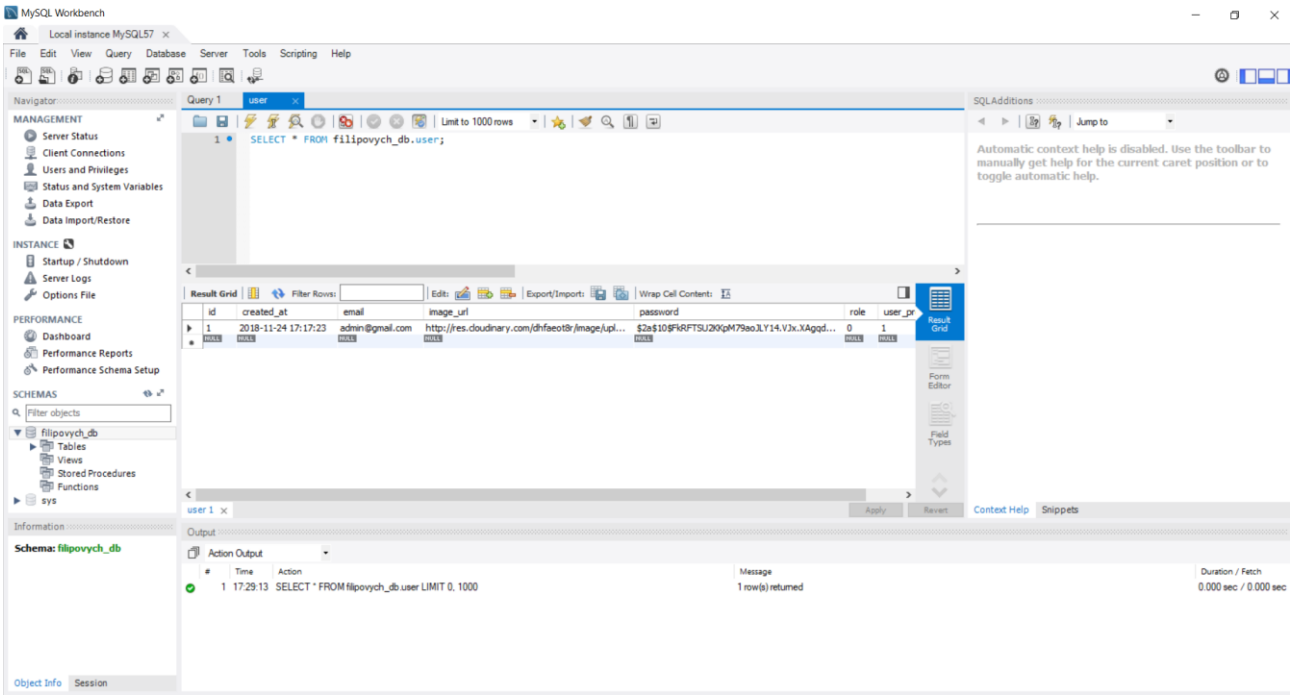

Рис.2. Таблиця в БД користувачів.

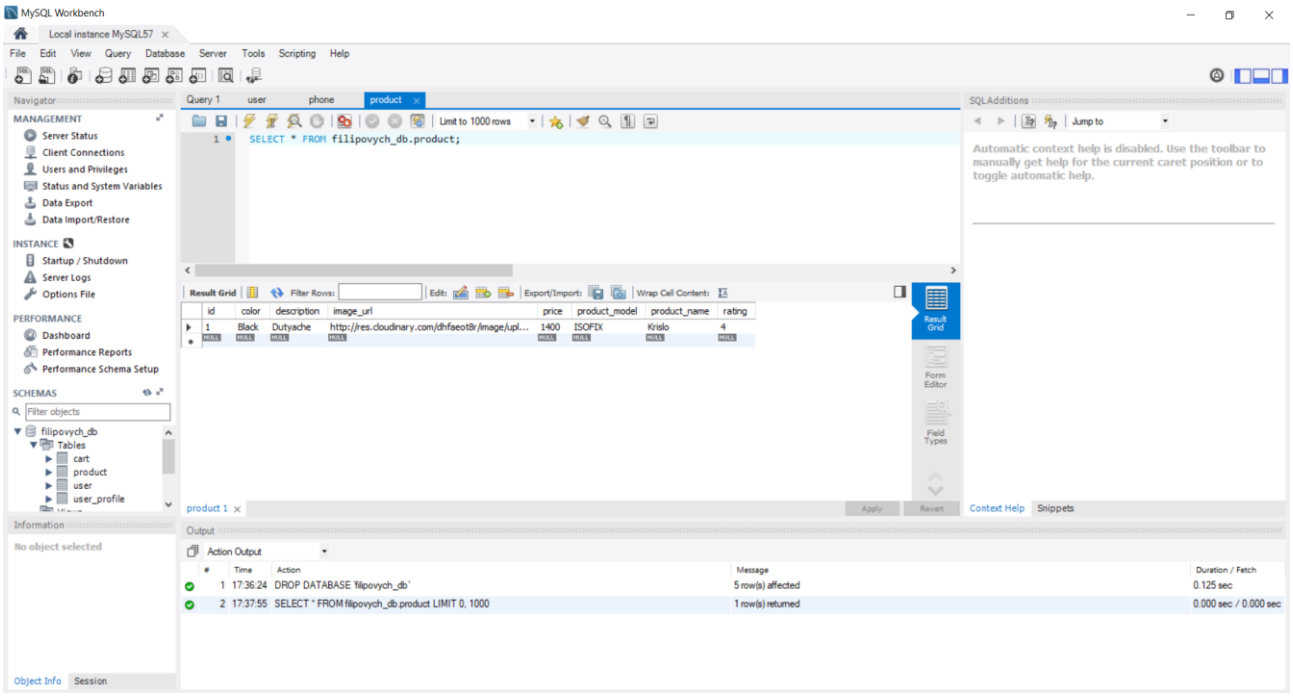

Рис.3. Таблиця в БД товарів.

Зчитування з БД:

Repository:

```
@Repository
public interface ProductRepository extends JpaRepository<Product, Integer> {

}
```

Service:

```
@Override
	public Page<Product> findProductByPage(Pageable pageable) {
		return productRepository.findAll(PageRequest.of(pageable.getPageNumber(),
		pageable.getPageSize(),
		pageable.getSort()));
```

Controller:

```
@GetMapping("/product-by-page")
	private String showProductsByPage(
			Model model,
			@PageableDefault Pageable pageable){
		Page<Product> page = productService.findProductsByPage(pageable);

		int currentPage = page.getNumber();
		int begin = Math.max(1, currentPage - 5);
		int end = Math.min(begin + 5, page.getNumber());

		model.addAttribute("beginIndex", begin);
		model.addAttribute("endIndex", end);
		model.addAttribute("currentIndex", currentPage);

		model.addAttribute("productListPageSize", page.getContent());

		return"admin/product-by-page";
}
```

Запис в БД:

Repository:

```
@Repository
public interface ProductRepository extends JpaRepository<Product, Integer> {

}
```

Service:

```
@Autowired private ProductRepository productRepository;

	@Override
	public void savePhone(Product phone) {
		productRepository.save(phone);
```

Controller:

```
@PostMapping("/add-product")
	public ModelAndView saveProduct(
			@Valid
			@ModelAttribute("productModelControl") Product product,
			@RequestParam("productImage") MultipartFile file,
			BindingResult br) {

		if (br.hasErrors() || file.getOriginalFilename().length() <= 0) {
			return new ModelAndView("admin/add-product","error","Check the
correct data entry");
		}

		try {
			String imageUrl = cloudinaryService.uploadFile(file, "product/" +
product.getId());
			product.setImageUrl(imageUrl);

			productService.saveProduct(product);
		} catch (Exception e) {
			return new ModelAndView("admin/add-product","error","Check the
correct data entry");
		}

		return new ModelAndView("redirect:/admin/products-by-page");
}
```

Висновок: На цій лабораторній роботі я розробив програму – сервер, яка буде проводити обробку запитів від програми – клієнта по протоколу TCP, записувати і зчитувати дані з бази даних.