**Student’s Individual Study #3**

**Full Name:** Bekenov Temirlan  
**Educational Program:** Faculty of IT

**Topic: Installing Packages**

**Target**

The purpose of this study is to prepare the environment for the TaskBoard project by installing all necessary packages and dependencies.  
This includes system-level software, programming libraries, and supporting services required for proper operation of the application and its components.  
Additionally, firewall rules will be configured to secure network access, and the installed packages will be verified through testing.

**Tasks**

* Define all required packages for the project;
* Install or update packages on the system;
* Configure firewall rules for secure access;
* Test installed packages and verify functionality.

**1. Required Packages**

For the TaskBoard project, which consists of a web server, backend API, and database, the following packages are required on the corresponding machines:

| **Machine** | **Component** | **Package/Library** | **Purpose** |
| --- | --- | --- | --- |
| VM1 | System utilities | curl, wget, git, ufw | Basic system operations and firewall management |
| VM1 | Web server | nginx | Reverse proxy and static file serving |
| VM1 | Backend | nodejs, npm | Running Node.js backend and installing dependencies |
| VM1 | Backend libraries | Express, dotenv, pg, cors (via npm) | Framework and libraries for backend development |
| VM2 | Database | postgresql, postgresql-contrib | Relational database management |
| VM2 | Database tools | net-tools, ufw | Connection management and firewall setup |

All packages were selected for compatibility with **Ubuntu Server 22.04 LTS**, and their installation was automated through shell scripts and package managers such as apt and npm.

**2. Installing Packages**

All packages were installed using the default Ubuntu package manager apt and the Node.js package manager npm.  
The installation process ensured that the latest stable versions were applied to maintain security and compatibility.

System packages were updated before installation to avoid dependency conflicts.  
This setup process was fully automated to reduce manual configuration time and maintain consistency across environments.

Each package installation step was verified by checking version outputs (e.g., nginx -v, node -v, psql --version).

**3. Firewall Configuration**

After installation, the **Uncomplicated Firewall (UFW)** was configured on both virtual machines to restrict unauthorized access.  
Firewall rules were applied to only allow required services:

| **Service** | **Port** | **Action** | **Description** |
| --- | --- | --- | --- |
| SSH | 22 | Allow | Remote administration and automation |
| HTTP | 80 | Allow | Web access for the TaskBoard frontend |
| HTTPS | 443 | Allow | Secure web access |
| PostgreSQL | 5432 | Allow (internal only) | Database connection within VirtualBox network |

UFW was enabled and tested to ensure only authorized connections were permitted.  
This configuration minimizes exposure of critical services and strengthens system security.

**4. Testing Installed Packages**

To verify successful installation and correct configuration of packages, several basic “smoke tests” were performed.

**Test examples:**

* Accessing the Nginx welcome page via browser to confirm the web server is running.
* Running a sample Node.js script to check server startup without errors.
* Connecting to PostgreSQL using a test user to verify database availability.

Each test confirmed that the installed components were operational and properly communicating within the network.

All tests were executed locally within the VirtualBox internal network and logged for documentation.

**5. Step-by-Step Report Summary**

1. Defined all necessary packages and libraries for the project environment.
2. Installed and updated system and application-level packages using apt and npm.
3. Configured and tested firewall rules using UFW for SSH, HTTP, HTTPS, and database access.
4. Performed smoke tests to verify the operation of installed packages and services.
5. Confirmed full functionality and security compliance of the prepared environment.

**Conclusions**

In this study, the project environment for the TaskBoard system was prepared by installing all required packages and dependencies.  
Firewall rules were established to protect the system while maintaining necessary connectivity between components.  
Testing verified that all services were correctly installed and operational.  
This environment now serves as a reliable foundation for further application development and deployment.