**Лабораторна робота № 5  
Кошти для виконання збірки проекту**

Виконав студент групи **КН-19-2 Ільченко Дмитро**

**Мета роботи:** отримати навички роботи із засобом виконання складання проекту.

**Хід виконання роботи**

1. Вивчити теоретичні відомості.

2. Створити файл для складання проекту свого індивідуального завдання.

3. Зробити висновки за результатами виконання роботи.

4. Виконати звіт про виконану роботі

Демонстрація файлу складання Electron-проекту (package.json):

{

  "name": "fitness-app",

  "productName": "fitness-app",

  "version": "1.0.0",

  "description": "My Electron application description",

  "main": ".webpack/main",

  "scripts": {

    "start": "electron-forge start",

    "package": "electron-forge package",

    "make": "electron-forge make",

    "publish": "electron-forge publish",

    "lint": "eslint --ext .ts,.tsx .",

    "test": "mocha -r ts-node/register src/test/test.ts"

  },

  "keywords": ["fitness","workout","diet","exercises","fitness app"],

  "author": {

    "name": "Exclusive-PG",

    "email": "daymerexc@gmail.com"

  },

  "license": "MIT",

  "config": {

    "forge": {

      "packagerConfig": {},

      "makers": [

        {

          "name": "@electron-forge/maker-squirrel",

          "config": {

            "name": "fitness\_app"

          }

        },

        {

          "name": "@electron-forge/maker-zip",

          "platforms": [

            "darwin"

          ]

        },

        {

          "name": "@electron-forge/maker-deb",

          "config": {}

        },

        {

          "name": "@electron-forge/maker-rpm",

          "config": {}

        }

      ],

      "plugins": [

        [

          "@electron-forge/plugin-webpack",

          {

            "devContentSecurityPolicy": "style-src \* self 'unsafe-inline' blob: data: gap:; script-src \* 'self' 'unsafe-eval' 'unsafe-inline' blob: data: gap:;",

            "mainConfig": "./webpack.main.config.js",

            "renderer": {

              "nodeIntegration": true,

              "webSecurity": false,

              "enableRemoteModule": true,

              "config": "./webpack.renderer.config.js",

              "entryPoints": [

                {

                  "html": "./src/index.html",

                  "js": "./src/renderer.ts",

                  "images": "./src/assets/images",

                  "name": "main\_window"

                }

              ]

            }

          }

        ]

      ]

    }

  },

  "devDependencies": {

    "@electron-forge/cli": "^6.0.0-beta.61",

    "@electron-forge/maker-deb": "^6.0.0-beta.61",

    "@electron-forge/maker-rpm": "^6.0.0-beta.61",

    "@electron-forge/maker-squirrel": "^6.0.0-beta.61",

    "@electron-forge/maker-zip": "^6.0.0-beta.61",

    "@electron-forge/plugin-webpack": "^6.0.0-beta.61",

    "@types/chai": "^4.3.3",

    "@types/jest": "^29.0.3",

    "@types/mocha": "^9.1.1",

    "@typescript-eslint/eslint-plugin": "^4.33.0",

    "@typescript-eslint/parser": "^4.33.0",

    "@vercel/webpack-asset-relocator-loader": "^1.7.0",

    "chai": "^4.3.6",

    "css-loader": "^6.7.1",

    "electron": "20.1.4",

    "eslint": "^7.32.0",

    "eslint-plugin-import": "^2.26.0",

    "file-loader": "^6.2.0",

    "fork-ts-checker-webpack-plugin": "^6.5.2",

    "mocha": "^10.0.0",

    "node-loader": "^2.0.0",

    "node-sass": "^7.0.3",

    "sass-loader": "^13.0.2",

    "style-loader": "^3.3.1",

    "ts-loader": "^9.3.1",

    "ts-node": "^10.9.1",

    "typescript": "^4.8.3"

  },

  "dependencies": {

    "@fortawesome/fontawesome-free": "^6.2.0",

    "axios": "^0.27.2",

    "electron-squirrel-startup": "^1.0.0",

    "swiper": "^8.4.2"

  }

}

Package.json є файлом-маніфестом для проекту. Він дає розробнику багато різнопланових можливостей. Наприклад, він являє собою центральний репозиторій налаштувань інструментальних засобів, що використовуються в проекті. Крім того, він є тим місцем, куди npm записує відомості про імена та версії встановлених пакетів.

В даному файлі можна назвати такі характеристики:

name - задає ім'я програми (пакета).

version — містить відомості про поточну версію програми.

description - короткий опис програми.

main — задає точку входу до програми.

private — якщо ця властивість встановлена ​​в true, це дозволяє запобігти випадковій публікації пакета в npm.

scripts - задає набір Node.js-скриптів, які можна запускати.

dependencies – містить список npm-пакетів, від яких залежить програма.

devDependencies - містить список npm-пакетів, що використовуються при розробці проекту, але не при його реальній роботі.

engines - задає список версій Node.js, на яких працює програма.

У вкладці config ми можемо налаштувати makers(збирачі проектів) та plugins(у нашому випадку webpack, налаштовує вхідну точку main і renderer процесів, а також систему безпеки проекту модулів)

Всі ці характеристики застосовуються або npm або іншими інструментальними засобами, що застосовуються протягом життєвого циклу застосування.

**Упаковка додатку Electron**

Існує 3 різних інструменти для пакування програми Electron: електрон-форж, електрон-пакувальник, електрон-білдер. Я використовую electron-forge на базі maker-squirrel. Після налаштування ми можемо запустити npm run make для створення пакету Windows та zip-файлу. Однак він не створюватиме пакети для Mac або Linux.

Electron створює пакети тільки для вашої поточної платформи, тому він не працює для пакування крос-платформних додатків, якщо ви не володієте кожною з платформ і не заперечуєте проти упаковки для кожної з них окремо.

**Висновок:**

Я отримав навички роботи із засобом виконання складання проекту.