Student’s name and surname:

Course: 4

ID: 210107111

Department: 6B06102-Computer Science,

Type of practice: Industrial practice 3 (CSS 480)

Me,  **Kabylbek Khamza**  confirmed that the internship period was completed in accordance with the work schedule of the internship from 25.07.2024 to 04.08.2024.

Company name:

Table:

|  |  |  |  |
| --- | --- | --- | --- |
| № | List of works to be performed (training) according to the program of diploma experience | Timeline of Industrial Internship | |
| start | finish |
| 1 | **Task 1** Install and configure all the necessary tools (Xcode, Git, CocoaPods).  **Task 2:** Examine the existing design documentation (if available) and the architecture of the application.  **Task 3:** Hold a meeting with the mentor to discuss the goals of the internship and identify key tasks. | 25.07.2024 | 26.07.2024 |
| 2 | **Task 1:** Learn the basics of Swift in depth, with an emphasis on memory management, multitasking using GCD and error handling.  **Task 2:** Design the application structure and define key modules, including ViewControllers and Services.  **Task 3:** Prepare documentation on the application architecture. | 26.07.2024 | 27.07.2024 |
| 3 | **Task 1:** Determine the data requirements and select the appropriate storage method (for example, Core Data, Realm).  **Task 2:** Design a data model, create entities and establish relationships between them.  **Task 3:** Write code to interact with the data storage layer (CRUD operations) and develop tests for these operations. | 27.07.2024 | 29.07.2024 |
| 4 | **Task 1:** Choose a framework or methodology for building a user interface (for example, UIKit, SwiftUI).  **Task 2:** Implement the main components of the interface, including handling user interactions and error management.  **Task 3:** Document the interface using tools such as Sketch or similar. | 29.07.2024 | 30.07.2024 |
| 5 | **Task 1:** Implement JWT-based user authentication to protect the API.  **Task 2:** Configure role-based access control (RBAC) for different API endpoints.  **Task 3:** Conduct security testing and vulnerability assessment. | 30.07.2024 | 31.07.2024 |
| 6 | **Task 1:** Write modular, integration and performance tests.  **Task 2:** Set up CI/CD pipelines for automated testing and deployment.  **Task 3:** Conduct a code review with the team to improve the quality of the code. | 31.07.2024 | 01.08.2024 |
| 7 | **Task 1:** Create a Dockerfile to containerize the application.  **Task 2:** Configure Docker Compose to manage services and their dependencies.  **Task 3:** Test the containerized application in different environments. | 01.08.2024 | 02.08.2024 |
| 8 | **Task 1:** Complete the final build of the application and prepare it for deployment.  **Task 2:** Complete the documentation, including deployment instructions and API usage guides.  **Task 3:** Prepare the final report on the completed work. | 02.08.2024 | 03.08.2024 |
| 9 | **Task 1:** Deploy the application in a test or staging environment.  **Task 2:** Perform load testing and monitoring of application performance.  **Task 3:** Refactor the code based on the test results. | 03.08.2024 | 04.08.2024 |
| 10 | **Task 1:** Present the completed project to the team or stakeholders.  **Task 2:** Transfer all source code, documentation and complete all administrative procedures.  **Task 3:** Summarize the internship and discuss further career steps or possible improvements. | 04.08.2024 | 04.08.2024 |

Supervisor’s grade: \_\_/60

Recommendations of the student:

**Focus on Continuous Learning:**

* Explore more advanced topics in Swift, such as asynchronous programming using async/await and performance optimization for large-scale applications. Continuous learning will help you deepen your knowledge and prepare you for more complex tasks in iOS development.

**Strengthen Understanding of Microservices:**

* In light of the trend towards using modular architecture, it will be useful to delve into the design and implementation of modular applications. This includes learning topics such as state management, inter-module interaction, and dependency coordination.

**Enhance Testing Skills:**

* Invest time in mastering testing methodologies, including writing comprehensive unit tests, integration tests, and performance tests. Understanding and applying the test-driven development (TDD) and behavior-driven development (BDD) approaches will significantly improve the quality of your code.