Experiment – 11

1. Aim: Comparative Study of Programming Languages for Web Development.

2. Objective: The aim of this laboratory experiment is to compare and evaluate the performance and efficiency of different programming languages commonly used in web development. The experiment will focus on the development of a simple web application using three different programming languages: Python (with Django framework), JavaScript (with Node.js and Express.js), and HTML-CSS. The study will analyze various aspects, including development speed, code readability, performance, and ease of implementation.

3. Experiment Setup:
Participants: A group of students with prior programming knowledge, including familiarity with Python, JavaScript, and HTML.

4. Equipment:
* Computers with the necessary development environments and web browsers.
* Text editors or Integrated Development Environments (IDEs) for coding.
* Internet connectivity for accessing documentation and resources.

5. Tasks:

a. Pre-Experiment Setup:
Participants will be divided into three groups, ensuring each group has members with expertise in one of the languages.

b. Experiment Execution:
* Each group is assigned the task of building a simple web application with similar functionality, such as a to-do list or a blog site.
* The Python group will use Django framework, the JavaScript group will use Node.js and Express.js, and the Ruby group will use Ruby on Rails.
* Participants will have a fixed amount of time to complete the development of their respective web applications.

c. Metrics and Evaluation: Case study will be evaluated based on following factors
* Measure the development time to complete the assigned task.
* Evaluate the readability and maintainability of the code in each language.
* Conduct performance tests to compare the speed & responsiveness of web applications.

d. Data Analysis:
* Analyze the data collected in development process and performance tests.
* Identify strengths and weaknesses of each programming language in the context of web development.

6 Discussion and Conclusion:
Share and discuss the findings.

References:
1. A case study on refactoring in Haskell programs