

## Project Name: Bank-Management System Using C++

### Implementation Milestones Plan (4 Weeks)

| Week   | Milestones/Goals                                      | Tasks/Activities  | Expected Outcomes  |
|--------|---|---|--|
| Week 1 | <b>Project Initialization and Environment Setup</b>   | <ul style="list-style-type: none"><li>- Define project requirements and features</li><li>- Set up a C++ development environment (IDE, compilers)</li><li>- Design project structure (classes, methods, etc.)</li></ul>  | A well-organized project plan, a working development setup, and a basic framework for the project.       |
| Week 2 | <b>Core Functionalities: Account Management</b>       | <ul style="list-style-type: none"><li>- Develop Bank Account class with attributes (account number, holder name, balance, type)</li><li>- Implement methods for account creation, modification, and reporting.</li><li>- Conduct unit tests for individual functionalities.</li></ul> | A functional system capable of creating, modifying, and displaying account details.                      |
| Week 3 | <b>Transaction Handling and Persistent Storage</b>    | <ul style="list-style-type: none"><li>- Add deposit and withdrawal functionality.</li><li>- Implement file handling for saving/loading account data (accounts.txt).</li><li>- Test transactions and data persistence.</li></ul>   | A smooth transaction system and persistent account storage that retains data between program executions. |
| Week 4 | <b>Search, Account Closure, and Final Refinements</b> | <ul style="list-style-type: none"><li>- Implement search and account closure functionality.</li><li>- Refactor code for modularity and readability.</li><li>- Conduct final testing and debugging.</li></ul>  | A fully functional console-based Bank Management System with modular, maintainable, and robust code.     |