

# Bank Loan

# Calculator –

# Project Report

**Submitted By:**

Ratnesha Gupta (590023288)

**Course:**

btech cse

**batch:**

45

**Date:**

28/11/25

## 2. Abstract

This project presents a menu-driven **Bank Loan Calculator** implemented in the C programming language. The system computes the Equated Monthly Installment (EMI) for loans, generates loan summaries, previews amortization schedules, and exports full repayment schedules to a CSV file. The project demonstrates structured programming principles, modular design, input validation, file handling, and mathematical computation of loan metrics. This report outlines the problem addressed, system design, implementation, testing, and potential future enhancements.

---

## 3. Problem Definition

Banks and financial institutions often require tools to compute loan repayments and EMI values quickly and accurately. Borrowers also benefit from understanding repayment structures before taking a loan. However, many manually performed calculations are time-consuming and prone to errors.

The problem addressed in this project is the lack of a simple, command-line based, C-program tool that:

1. Computes EMI using standard formulas.
2. Provides loan summaries including total interest and total repayment.
3. Generates amortization schedules.
4. Allows exporting the schedule to an external CSV file.
5. Handles improper or invalid user input.

The objective is to create a robust, modular, and user-friendly loan calculator for educational and practical use.

# 4. System Design

## Algorithm

**Step 1:** Start program

**Step 2:** Display menu (Detailed flow, Quick EMI, Exit)

**Step 3:** If Detailed Flow:

- Read principal
- Read annual interest rate
- Read months or years (convert to months)
- Compute EMI
- Display summary
- Ask if user wants preview (first 12 months)
- Ask if user wants to save CSV

**Step 4:** If Quick EMI:

- Read principal, interest rate, tenure (years)
- Convert years to months
- Compute EMI
- Display EMI

**Step 5:** If Exit: Terminate

**Step 6:** Return to menu

# 5. Implementation Details

The project is implemented in C using standard libraries:

stdio.h, stdlib.h, string.h, math.h.

## 5.1 EMI Calculation Function

```
double compute_emi(double principal, double annual_rate_percent, int months) {  
    double monthly = annual_rate_percent / 100.0 / 12.0;  
    if (monthly == 0.0) return principal / (double)months;  
    double r = monthly;  
    double powv = pow(1.0 + r, months);  
    double emi = principal * r * powv / (powv - 1.0);  
    return emi;  
}
```

## 5.2 Summary Display

```
void show_summary(double principal, double annual_rate_percent, int months) {
    double emi = compute_emi(principal, annual_rate_percent, months);
    double total = emi * months;
    double interest = total - principal;
    printf("\n--- Loan Summary ---\n");
    printf("Principal      : %.2f\n", principal);
    printf("Annual Rate %% : %.2f\n", annual_rate_percent);
    printf("Tenure months : %d\n", months);
    printf("Monthly EMI   : %.2f\n", emi);
    printf("Total payment : %.2f\n", total);
    printf("Total interest: %.2f\n", interest);
}
```

## 5.3 CSV Export Function

```
if(choice == 2 && *fifty){
    *fifty = 0;
    int wrongs[3], idx = 0;
    for(int i = 0; i < 4; i++)
        if(order[i] != q->correct)
            wrongs[idx++] = i;

void save_schedule(double principal, double annual_rate_percent, int months,
const char *fname) {
    FILE *f = fopen(fname, "w");
    if (!f) { perror("open"); return; }

    double balance = principal;
    double monthly = annual_rate_percent / 100.0 / 12.0;
    double emi = compute_emi(principal, annual_rate_percent, months);

    fprintf(f, "Month,EMI,Interest,Principal,Balance\n");

    for (int m = 1; m <= months; ++m) {
        double interest = balance * monthly;
        double principal_paid = emi - interest;
        if (m == months) { principal_paid = balance; emi = interest +
principal_paid; }
        balance -= principal_paid;

        fprintf(f, "%d,%2f,%2f,%2f,%2f\n",

```

```
    m, emi, interest, principal_paid, balance);  
}  
fclose(f);  
}
```

## 6. Testing & Results

### Test Case 1: Valid Input

- Principal: 500000
- Annual Rate: 6.5%
- Tenure: 20 years (240 months)

### Output:

- Monthly EMI: 3723.00
- Total Payment: 893,520.00
- Total Interest: 393,520.00
- Preview shows first 12 months correctly
- CSV file successfully generated

### Test Case 2: Zero Interest

- Principal: 100000
- Rate: 0%
- Tenure: 10 months

### Output:

- EMI: 10,000
- No compounding applied
- Amortization reduces linearly

### Test Case 3: Invalid Inputs

- Negative principal
- Negative rate
- Non-numeric input

## **Result:**

Program displays appropriate error messages and reprompts for valid input.

## **Observation**

The program behaves reliably under normal and edge cases. CSV output imports correctly in Excel.

## **7. Conclusion & Future Work**

### **Conclusion**

The Bank Loan Calculator successfully computes EMI, displays detailed summaries, previews amortization schedules, and exports data to CSV files. The design is modular, user-friendly, and demonstrates effective C programming concepts including input handling, mathematical computation, and file I/O.

### **Future Enhancements**

- Add graphical interface (GUI version).
- Support for multiple repayment types (reducing balance, fixed, balloon payments).
- Include loan comparison between two or more banks.
- Add PDF export functionality.
- Save and load previous loan configurations.

## **8. References**

1. Balagurusamy, E. *Programming in ANSI C*, McGraw Hill.
2. Kernighan, B., Ritchie, D. *The C Programming Language*.
3. Standard EMI Formula – Reserve Bank of India Financial Guidelines.
4. C Standard Library Documentation – stdio.h, stdlib.h, math.h.