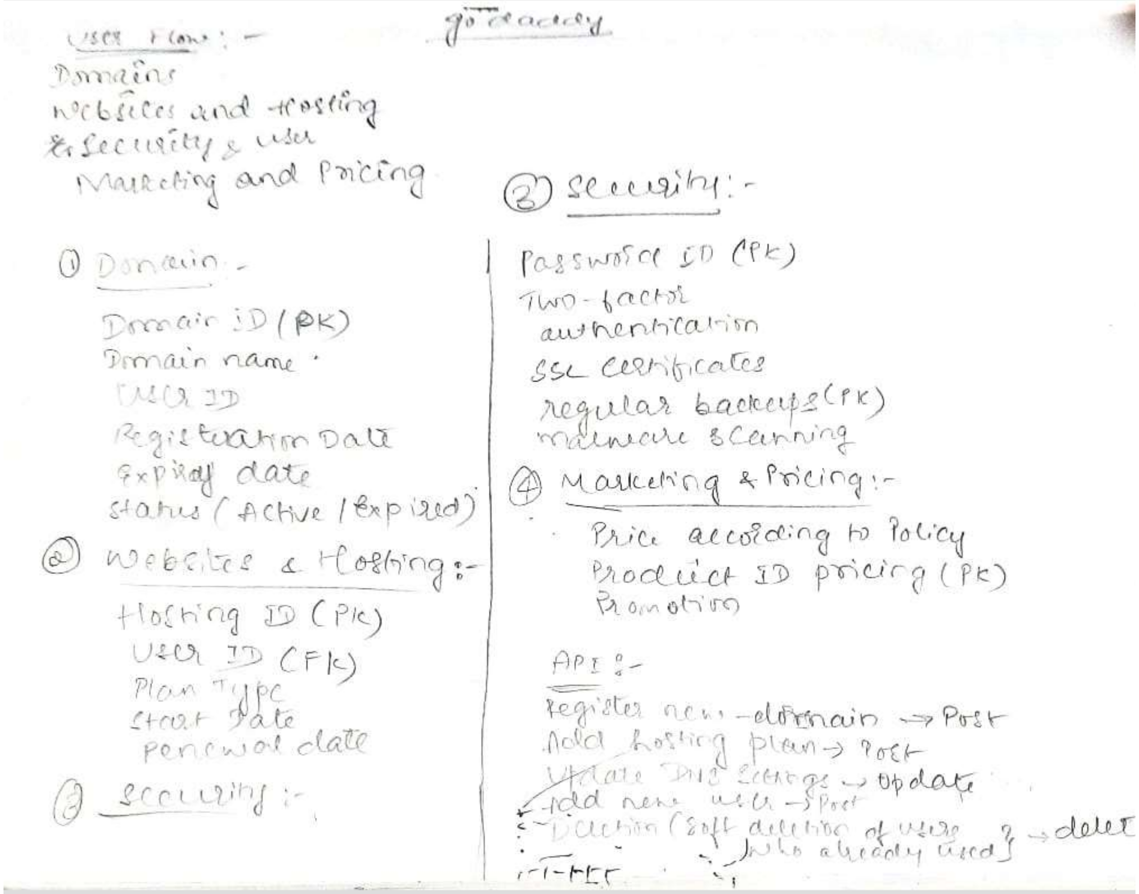
Assesment-5(Friday)

# Go-daddy Problem solving:

Overview:



godaddy

User Flow :-
Domains
Websites and Hosting
& Security & user
Marketing and Pricing

① Domain -

Domain ID (PK)
Domain name .
User ID
Registration Date
Expiry date
Status (Active / Expired)

② Websites & Hosting :-

Hosting ID (PK)
User ID (FK)
Plan Type
Start Date
Renewal date

③ Security :-

③ Security :-

Password ID (PK)
Two-factor authentication
SSL certificates
regular backups (PK)
malware scanning

④ Marketing & Pricing :-

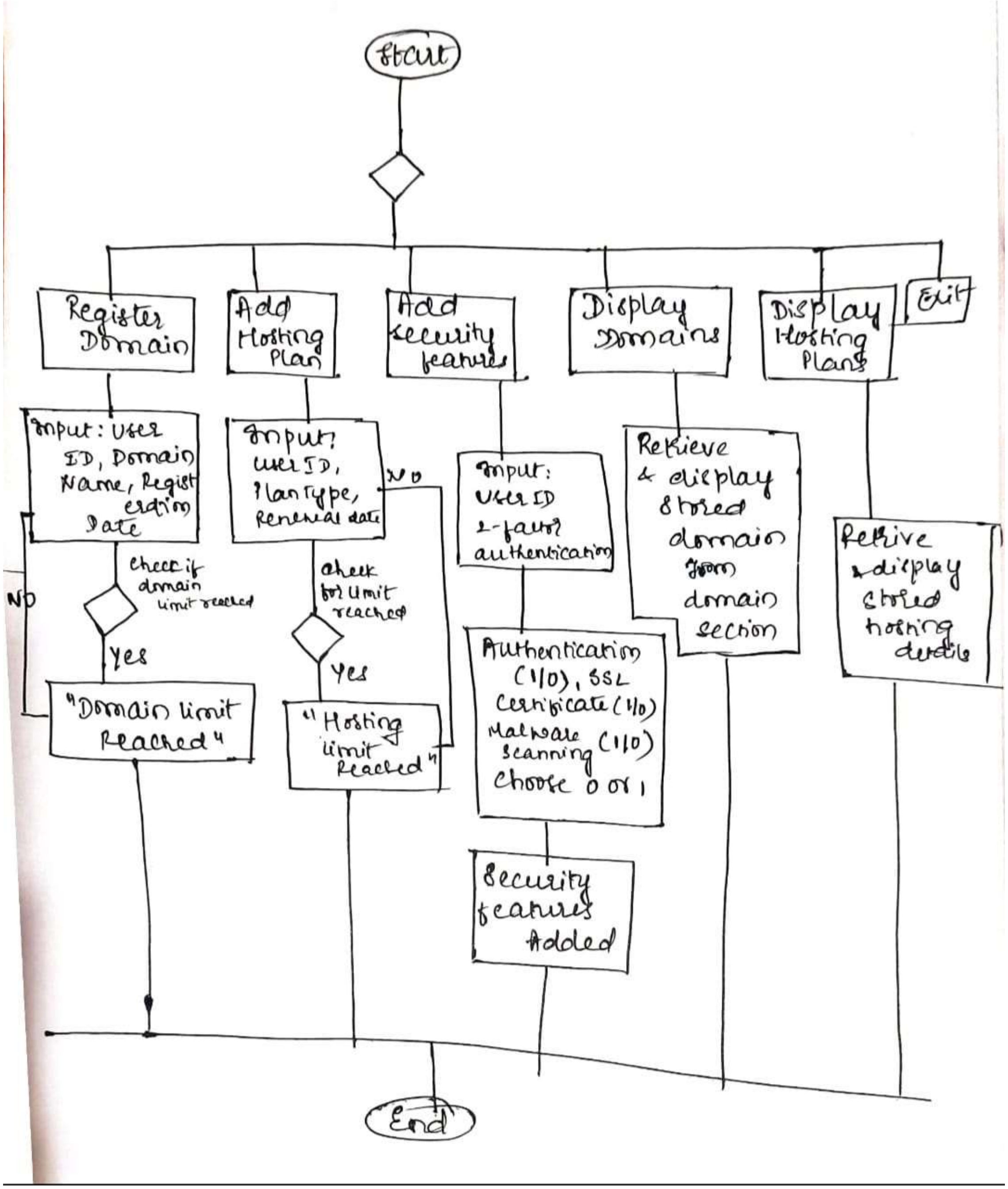
Price according to Policy
Product ID pricing (PK)
Promotion

API :-

Register new-domain → Post
Add hosting plan → Post
Update DNS settings → Update
Add new user → Post
Deletion (soft deletion of users who already used) → delete

Assesment-5(Friday)

FLOWCHART:


Start
Register Domain
Add Hosting Plan
Add Security features
Display Domains
Display Hosting Plans
Exit
Input: User ID, Domain Name, Registeration Date
check if domain limit reached
NO
Yes
"Domain limit Reached"
Input: User ID, Plan Type, Renewal date
check for limit reached
NO
Yes
"Hosting limit Reached"
Input: User ID 2-factor authentication
Authentication (1/0), SSL Certificate (1/0) Malware scanning (1/0) Choose 0 or 1
Security features Added
Retrieve & display stored domain from domain section
Retrive & display stored hosting details
End

Assesment-5(Friday)

CODE:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_DOMAINS 100
#define MAX_HOSTINGS 100
#define MAX_USERS 100

typedef struct {
    int domain_id;
    char domain_name[100];
    int user_id;
    char registration_date[11];
    char status[10];
} Domain;

typedef struct {
    int hosting_id;
    int user_id;
    char plan_type[50];
    char renewal_date[11];
} Hosting;

typedef struct {
    int password_id;
    int user_id;

    int two_factor_auth;

    int ssl_certification;

    int malware_scanning;

} Security;


Domain domains[MAX_DOMAINS];

Hosting hostings[MAX_HOSTINGS];

Security securities[MAX_USERS];


int domain_count = 0;

int hosting_count = 0;

int user_count = 0;


void register_domain(int user_id, const char* domain_name, const char*
registration_date) {

    if (domain_count < MAX_DOMAINS) {

        domains[domain_count].domain_id = domain_count + 1;

        strcpy(domains[domain_count].domain_name, domain_name);

        domains[domain_count].user_id = user_id;

        strcpy(domains[domain_count].registration_date, registration_date);

        strcpy(domains[domain_count].status, "active");

        domain_count++;

        printf("Domain registered: %s\n", domain_name);

    } else {

        printf("Domain limit reached.\n");

    }
```

```
}


void add_hosting_plan(int user_id, const char* plan_type, const char*
renewal_date) {
    if (hosting_count < MAX_HOSTINGS) {
        hostings[hosting_count].hosting_id = hosting_count + 1;
        hostings[hosting_count].user_id = user_id;
        strcpy(hostings[hosting_count].plan_type, plan_type);
        strcpy(hostings[hosting_count].renewal_date, renewal_date);
        hosting_count++;
        printf("Hosting plan added: %s\n", plan_type);
    } else {
        printf("Hosting limit reached.\n");
    }
}


void add_security_features(int user_id, int two_factor_auth, int
ssl_certification, int malware_scanning) {
    if (user_count < MAX_USERS) {
        securities[user_count].password_id = user_count + 1; // Simple ID
assignment
        securities[user_count].user_id = user_id;
        securities[user_count].two_factor_auth = two_factor_auth;
        securities[user_count].ssl_certification = ssl_certification;
        securities[user_count].malware_scanning = malware_scanning;
        user_count++;
        printf("Security features added for user ID: %d\n", user_id);
    } else {
        printf("User limit reached.\n");
    }
}

void display_domains() {
    printf("\nRegistered Domains:\n");
    for (int i = 0; i < domain_count; i++) {
        printf("ID: %d, Name: %s, User ID: %d, Registration Date: %s, Status: %s\n",
               domains[i].domain_id, domains[i].domain_name, domains[i].user_id,
               domains[i].registration_date, domains[i].status);
    }
}

void display_hostings() {
    printf("\nHosting Plans:\n");
    for (int i = 0; i < hosting_count; i++) {
        printf("ID: %d, User ID: %d, Plan Type: %s, Renewal Date: %s\n",
               hostings[i].hosting_id, hostings[i].user_id,
               hostings[i].plan_type, hostings[i].renewal_date);
    }
}

int main() {
    int ch;
    Domain a;
```

Assesment-5(Friday)

```
    Hosting b;
    Security c;

    while (1) {
        printf("\n1. Register Domain\n2. Add Hosting Plan\n3. Add Security Features\n4. Display Domains\n5. Display Hosting Plans\n6. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &ch);

        switch (ch) {
            case 1:
                printf("Enter User ID, Domain Name, Registration Date (YYYY-MM-DD): ");
                scanf("%d %s %s", &a.user_id, a.domain_name, a.registration_date);
                register_domain(a.user_id, a.domain_name, a.registration_date);
                break;

            case 2:
                printf("Enter User ID, Plan Type, Renewal Date (YYYY-MM-DD): ");
                scanf("%d %s %s", &b.user_id, b.plan_type, b.renewal_date);
                add_hosting_plan(b.user_id, b.plan_type, b.renewal_date);
                break;

            case 3:
                printf("Enter User ID, Two Factor Auth (1/0), SSL Certification (1/0), Malware Scanning (1/0): ");
                scanf("%d %d %d %d", &c.user_id, &c.two_factor_auth, &c.ssl_certification, &c.malware_scanning);
                add_security_features(c.user_id, c.two_factor_auth, c.ssl_certification, c.malware_scanning);
                break;

            case 4:
                display_domains();
                break;

            case 5:
                display_hostings();
                break;

            case 6:
                exit(0);

            default:
                printf("Invalid choice. Try again.\n");
        }
    }
}
```

OUTPUT:

Assesment-5(Friday)

```
1. Register Domain
2. Add Hosting Plan
3. Add Security Features
4. Display Domains
5. Display Hosting Plans
6. Exit
Enter your choice: 1
Enter User ID, Domain Name, Registration Date (YYYY-MM-DD):
023
java-programming
2025-02-19
Domain registered: java-programming

1. Register Domain
2. Add Hosting Plan
3. Add Security Features
4. Display Domains
5. Display Hosting Plans
6. Exit
Enter your choice: 2
Enter User ID, Plan Type, Renewal Date (YYYY-MM-DD):
023
premium
2025-02-27
Hosting plan added: premium
```

Assesment-5(Friday)

```
1. Register Domain
2. Add Hosting Plan
3. Add Security Features
4. Display Domains
5. Display Hosting Plans
6. Exit
Enter your choice: 3
Enter User ID, Two Factor Auth (1/0), SSL Certification (1/0), Malware Scanning (1/0): 23 1 1 1
Security features added for user ID: 23

1. Register Domain
2. Add Hosting Plan
3. Add Security Features
4. Display Domains
5. Display Hosting Plans
6. Exit
Enter your choice: 4

Registered Domains:
ID: 1, Name: java-programming, User ID: 23, Registration Date: 2025-02-19, Status: active

1. Register Domain
2. Add Hosting Plan
3. Add Security Features
4. Display Domains
```

```
5. Display Hosting Plans

6. Exit

Enter your choice: 5


Hosting Plans:

ID: 1, User ID: 23, Plan Type: premium, Renewal Date: 2025-02-27


1. Register Domain

2. Add Hosting Plan

3. Add Security Features

4. Display Domains

5. Display Hosting Plans

6. Exit

Enter your choice: 6
```