# 

# 

# MINI PROJECT

# ON

# PRISON MANAGEMENT SYSTEM

# 

# 

# 

# INDEX

**S. No Name of the Content Page No**

1. Problem Statement 1

2. Description 1

3. Features 2

4. Requirements 3

5. Test cases 4

6. Expected Results 5

**1. Problem Statement:**

To provide a system which is based on a concept to generate all the prisoner’s details such as prisoner’s id, name, age, gender, height, weight, crime, conviction etc.

**2. Project Description:**

This project titled “Prison Management System” is built in the view of providing management software for handling the inner working of any small jail across the country.

This system is based on a concept to generate all the prisoner’s record. From this system, the user can easily add each and every prisoner’s detail such as: prisoner’s id, name, age, gender, height, weight, crime, conviction etc. This user can view a list of the prisoner, search record and delete too. This system contains login system. Even, to remove a prisoner’s record user has to pass through password verification. So, there’s no chance of data loss or misuse and it is not time-consuming. The whole project is designed in ‘C’ Programming language & different variables and strings have been used for the development of this project. It’s easy to operate and understand by users.

This system will record the following details of the prisoner

PRISONER'S NAME IS:

PRISONER'S GENDER IS:

PRISONER'S AGE IS:

PRISONER'S WEIGHT IS:

PRISONER'S HEIGHT IS:

PRISONER'S HAIRCOLOR IS:

PRISONER'S EYECOLOR IS:

PRISONER'S CRIME IS:

THE PLACE WHERE THE PRISONER WAS CONVICTED IS:

COURT IS:

PRISONER'S LAWYER IS:

PRISONER'S CONVICTION IS:

THE DATE PUNISHMENT STARTED AT IS:

THE DATE PUNISHMENT ENDS AT IS:

PRISONER'S EMERGENCY CONTACT IS:

RELATION OF EMERGENCY CONTACT WITH PRISONER IS:

EMERGENCY CONTACT'S PHONE NUMBER IS:

**3. Features:**

* Login system with validations
* Focused on Security
* Each and Every detail of the prisoner
* Easy to operate by the users

**4. Requirements:**

**4.1 Functional Requirements**

1. Add new records

Admin should have the ability to add new records. Admin should add various information such as prisoner id, prisoner name, date of birth

1. Search, delete and view record

Admin should be able to search through the records using either name or prisoner id. Deletion of records can also be done using prisoner id. View of complete records should be provided to the admin.

1. View previous placed records

The admin should be able to view all the previously placed records for further usage.

**4.2 Non-Functional requirements**

1. Portability

System running on one platform can be easily converted to run on another platform

1. Reliability

The ability of the system to behave consistently in acceptable manner when operating within the environment for which the system was intended.

1. Availability

The system should be available at all times, meaning the user can access it all the time, only restricted by the down time of the server on which the system runs.

1. Security

Secure access of confidential data (prisoner information)

5. Performance

Performance should be fast.

6. Privacy

Personal data of the system should not disclose to anyone.

**5. TEST CASES**

#### TABLE 1: ADD NEW RECORD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **TEST SCENARIO** | **TEST STEP** | **EXPECTED RESULT** | **ACTUAL OUTPUT** |
| **1** | **Verify that correct option is selected from the menu** | **Selecting any option from the menu.** | **Go to that particular option.** | **The option selected gets displayed.** |
| **2** | **Verify if different option is selected apart from the menu options** | **Selecting NON menu option.** | **Should give an error** | **Displays saying option doesn’t exist and returns to the menu.** |

#### TABLE 2: SEARCH RECORD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **TEST SCENARIO** | **TEST STEP** | **EXPECTED RESULT** | **ACTUAL OUTPUT** |
| **1** | **Prisoner name: Verify if the given name is available** | **Entering the name.** | **Accepts the name and gives the output** | **Displays the particular information of the given name.** |
|  | **Verify if the given name is NOT available** | **Entering the name** | **Should return that it doesn’t exist.** | **Displays that it doesn’t exist and asks to re-enter the correct name.** |
| **2** | **Prisoner ID: Verify if the given ID is available** | **Entering the ID.** | **Accepts the ID and gives the output** | **Displays the particular information of the given ID.** |
|  | **Verify if the given ID is NOT available** | **Entering the ID** | **Should return that it doesn’t exist.** | **Displays that it doesn’t exist and asks to re-enter the correct ID.** |

#### TABLE 3: DELETE RECORD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SL NO** | **TEST SCENARIO** | **TEST STEP** | **EXPECTED RESULT** | **ACTUAL OUTPUT** |
| **1** | **Prisoner ID: Verify if the given ID is available** | **Entering the ID.** | **Accepts the ID and deletes the record** | **Displays the particular information of the given ID and deletes the record.** |
|  | **Verify if the given ID is NOT available** | **Entering the ID** | **Should return that it doesn’t exist.** | **Displays that it doesn’t exist and asks to re-enter the correct ID.** |