

# **GATE-PASS MANAGEMENT**

**LOCAL OD | VISITOR'S ENTRY**

## **PROJECT REPORT**



**Project by: Rohan Rahul Takmoge**  
Project-Trainee,  
Ashoka Buildcon Limited.

**Guide by: Kiran Chougule**  
Asst. General Manager-IT,  
Ashoka Buildcon Limited.

# CONTENTS

INTRODUCTION TO PROJECT .....	2
ABSTRACT .....	3
MODEL, VIEW AND CONTROLLER.....	4
DATABASE .....	9
SOFTWARE and HARDWARE REQUIREMENTS.....	10
SCREENSHOTS .....	11
CONCLUSION .....	16

# INTRODUCTION

The **Gate Pass Management System** is a web-based application that provides an interface for managing the entries and exits of visitors and employees in an organization.

The system has two main modules: Visitors Entry and Local OD.

The **Visitors Entry** module allows the entry of visitors into the organization and records the details of the Entry Date & Time, Visitor Mobile No. , Visitor Name, Company Name, Employee Name, Purpose of Visit, Visit Date and Time (System Date and Time when entered), Visit End Date & Time (System Date and Time while leaving), Vehicle Type & Vehicle No.

The **Local OD** module allows employees to apply for local out duty and records the details of the Employee No and Name, Visit Location, Purpose Of Visit (in the description), Type of Visit, Type of Local OD, Out Date and Time, In Date and Time.

A detailed Explanation of these Modules is Provided in the Model, View, and Controller Section

## Technology Used:

The Gate Pass Management System is built using ASP.NET MVC framework with Entity Framework Core and SQL Server database.

The front end of the application is designed using HTML, CSS, JavaScript and Bootstrap. The application uses Microsoft Excel Library (EPPlus) to export data to Excel.

# ABSTRACT

The Gate Pass Management System is an ASP.NET MVC web application designed to streamline and automate the process of managing gate passes for visitors and employees. The system provides an easy-to-use interface for managing the issuance of gate passes, tracking the movement of visitors and employees, and generating reports on gate pass activities.

The system is built using the ASP.NET MVC framework and is backed by a Microsoft SQL Server database. It uses Entity Framework to interact with the database and provides a rich set of features for managing the data. The system is designed to be modular and extensible, with separate modules for managing Local OD and Visitor Entry

The system is designed to be easy to use and intuitive, with a clean and modern interface that allows administrators to quickly perform the most common tasks.

Overall, the Gate Pass Management System provides an efficient and effective way to manage gate passes for visitors and employees, reducing the administrative burden and improving the accuracy and timeliness of gate pass data. The system is highly customizable and can be tailored to meet the specific needs of any organization, making it an ideal solution for facilities of all sizes.

# MVC

## MODEL

The **LocalOD** and **VisitorsEntry** are the two main models in the Gate Pass Management system, which is developed to manage and monitor the entry and exit of visitors and employees in an organization.

The **LocalOD** model is used to capture the details of employees who go out of the office premises for a particular purpose and return within the same day. It has **three types of LocalODs** – Since Morning LOD, Intermediate LOD, and Upto Evening LOD. The fields that need to be captured while creating a LocalOD are Employee No and Name, Visit Location, Purpose of Visit, Type of Visit, Type of LocalOD, Out Date and Time, and In Date and Time.

The **Employee No and Name** where employees who are authorized to go out of the office premises will enter his/her No and Name. The **Visit Location** field is there that contains the locations where an employee can go. The **Purpose of Visit** field is a textbox that allows the employee to enter the reason for going out in the description. The **Type of Visit** is a dropdown that contains two options 1. Official and 2. Personal employee has to select from these options. The **Type of LocalOD** field is a dropdown menu that contains the three types of LocalODs mentioned above.

The **Out Date and Time** field capture the date and time when an employee goes out of the office premises, and it is system-generated for Intermediate LOD and Upto Evening LOD. For Since Morning LOD, it will autofill time to 9:30 A.M. The **In Date and Time** field captures the date and time when an employee returns to the office premises, and it is system-generated for Intermediate LOD and Since Morning LOD and will update after employee click on punch. For Upto Evening LOD, it will autofill time to 18:30 P.M. which is office-end time.

The **VisitorsEntry** model is used to capture the details of visitors who come to the organization for a particular purpose. The fields that had been captured while creating a VisitorsEntry are Entry Date and Time, Visitor Mobile No., Visitor Name, Company Name, Employee Name, Purpose of Visit, Visit Date and Time, Visit End Date and Time, and Vehicle Type & Vehicle No.

The **Entry Date and Time** field is a system-generated field that captures the date and time when a visitor enters the details in the system. The **Visitor Mobile No.** field is a textbox that captures the mobile number of the visitor, the other fields such as **Visitor Name, Company Name, and Employee Name** are also textbox that asks the visitor to enter details. The **Purpose of Visit field** is a textbox that allows the visitor to enter the reason for the visit. The **Visit Date and Time** field is a system-generated field that captures the date and time when the visitor starts the visit. The Visit End Date and Time field is a system-generated field that captures the date and time when the visitor leaves the office premises by clicking on the punch option. The **Vehicle Type & Vehicle No.** field is a textbox that captures the type and number of the vehicle used by the visitor to come to the office.

Both models are designed to capture the relevant details of employees and visitors who go out or come to the organization's premises. By using these models, the organization can monitor and manage the entry and exit of employees and visitors. The captured data can be used to generate reports such as the number of visitors, the purpose of visits, the time spent by visitors, the number of LocalODs, etc. This information can be used to make informed decisions regarding the entry and exit policies of the organization.

## VIEW

In the project, various views have been used to render the HTML content of the application. These views are responsible for displaying the data and taking inputs from the user. Here is a detailed explanation of the views used in the project:

### 1. **Index** view:

This view is used to display the list of entries in the **LocalOD** and **VisitorsEntry** models. It provides options to create new entries, edit existing entries, and delete them. Additionally, it allows the user to export the data to an Excel file and also gives an option of searching the exact data using a name. The view uses a table to display the entries and provides links to create new entries and export data.

### 2. **Create** view:

This view is used to create a new entry in the **LocalOD** and **VisitorsEntry** models. It provides a form to enter the necessary details of Visitor in Visitors Entry and Employee in Local OD. After entering the details, the user can submit the form to create a new entry.

### 3. **Delete** view:

This view is used to confirm the deletion of an existing entry in the **LocalOD** and **VisitorsEntry** models. It displays the details of the entry that is being deleted and asks for confirmation before deleting the entry.

### 4. **Punch** view:

This view is used in **LocalOD** to update In Date and Time and in **VisitorsEntry** to update Visit End Date and Time by clicking on the Punch button available in the index view.

Overall, the views used in the project provide a user-friendly interface to create, edit, delete, search, and export the data in the LocalOD and VisitorsEntry models. They provide an easy and efficient way for the users to manage the data in the application.

# CONTROLLER

In the project, two controllers have been used: "LocalODController" and "VisitorsEntryController". Let's discuss both of these controllers in detail:

## 1. LocalODController:

The "LocalODController" is used to handle all the actions related to Local OD (Official Duty) entries. The following actions have been implemented in this controller:

- **Index:** This action method is responsible for fetching all the Local OD entries from the database and displaying them in the view.
- **Create:** This action method is used to display a view where the user can create a new Local OD entry. The view contains various fields such as Employee No/Name Selection, Visit Location, Purpose of Visit, Type of Local OD, Out Date and Time, In Date and Time, etc. Once the user submits the form, the entry is added to the database.
- **Delete:** This action method is used to delete an existing Local OD entry from the database.
- **ExportToExcel:** This action method is used to export all the Local OD entries to an Excel file. This method uses the "EPPlus" library to generate the Excel file.
- **Punch:** This action method is used to update In Date and Time in LocalOD.

## 2. **VisitorsEntryController:**

The "VisitorsEntryController" is used to handle all the actions related to visitor entries. The following actions have been implemented in this controller:

- **Index:** This action method is responsible for fetching all the visitor entries from the database and displaying them in the view.
- **Create:** This action method is used to display a view where the user can create a new visitor entry. The view contains various fields such as Visitor Name, Visitor Mobile No., Company Name, Employee Name, Purpose of Visit, Visit Date and Time, Visit End Date and Time, Vehicle Type, and Vehicle No. Once the user submits the form, the entry is added to the database.
- **Delete:** This action method is used to delete an existing visitor entry from the database.
- **ExportToExcel:** This action method is used to export all the visitor entries to an Excel file. This method uses the "EPPlus" library to generate the Excel file.
- **Punch:** This action method is used to update Visit End Date and Time in Visitor Entry.

Overall, these controllers help in managing and maintaining the Local OD and visitor entries in the system.

# DATABASE

The project uses a **SQL Server** database to store the information for the LocalOD and VisitorsEntry models. The database consists of two tables, one for each model.

The **AppDbContext** class sets up the database context and defines the **DbSet** properties for each table. It also defines the relationship between the two tables using the **OnModelCreating** method, which sets the foreign key for VisitorsEntry to LocalOD.

The **LocalOD** table has seven fields:

1. **Id** (int): a unique identifier for each LocalOD entry.
2. **EmployeeName** (String): Employee Name of employee going for LocalOD.
3. **VisitLocation** (String): Location the employee is visiting.
4. **PurposeOfVisit** (string): the reason for the employee's visit.
5. **TypeOfLocalOD** (String): the type of LocalOD requested (since morning, intermediate, or upto evening).
6. **OutDateTime** (String): the date and time the employee will leave for the visit.
7. **InDateTime** (String) : The date and time the employee is reached after visit.

The **VisitorsEntry** table has eleven fields:

1. **Id** (int): a unique identifier for each VisitorsEntry entry.
2. **EntryDateTime** (String): the date and time the visitor entered the facility.
3. **VisitorMobileNumber** (string): the mobile number of the visitor.
4. **VisitorName** (string): the name of the visitor.
5. **CompanyName** (string): the name of the company the visitor represents.
6. **EmployeeName** (string): the name of the employee the visitor is meeting.
7. **PurposeOfVisit** (string): the reason for the visitor's visit.
8. **VisitDateTime** (String): the date and time of the visit.
9. **VisitEndDateTime** (String): the date and time of the end of visit.
10. **VehicleType** (String): the type of the vehicle the visitor arrived in.
11. **VehicleNo** (String): the no. of the vehicle the visitor arrived in.

Overall, the database structure is simple and efficient, allowing for easy retrieval and management of the LocalOD and VisitorsEntry data.

# REQUIREMENTS

**Hardware** and **software** requirements to run this project can vary based on the size and complexity of the project.

However, here are some general recommendations:

## **Hardware Requirements:**

- Processor: 1.8 GHz or faster processor, dual-core or better
- RAM: 4 GB or more
- Storage: At least 10 GB of free disk space
- Display: Monitor with 1280x768 resolution or higher

## **Software Requirements:**

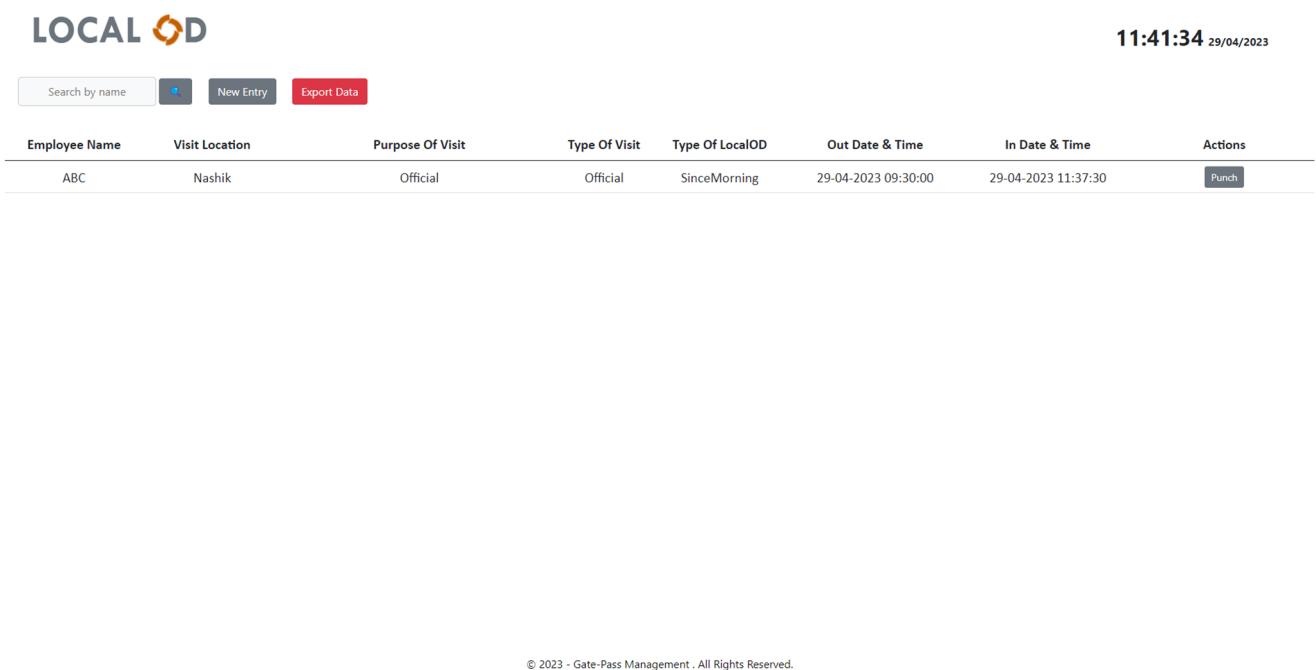
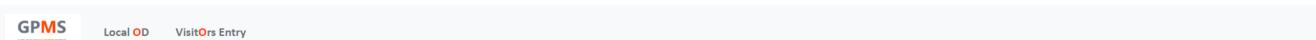
- Operating System: Windows 10 or higher, or any modern version of Linux or macOS
- Visual Studio Code or Visual Studio 2019 or higher
- .NET Core SDK 5.0 or higher
- Microsoft SQL Server 2017 or higher

## **Additional Requirements:**

- Basic knowledge of C# programming language and ASP.NET Core framework
- Basic knowledge of HTML, CSS, and JavaScript for frontend development
- Knowledge of Entity Framework Core and LINQ for working with databases
- Knowledge of Git and version control for collaboration and code management

It's important to note that these **requirements may change depending on the specific needs of the project**. It's always a good idea to consult the project requirements and consult with other developers and stakeholders to ensure that all necessary hardware and software are available and compatible.

# SCREENSHOTS



© 2023 - Gate-Pass Management . All Rights Reserved.

**CREATE NEW ENTRY**

---

Employee No. :

Employee Name :

Visit Location :

Purpose Of Visit :

Type Of Visit :  Personal

Type of LocalOD :  -- Select an option --

Out Date and Time :

In Date and Time :

Create Cancel

© 2023 - Gate-Pass Management . All Rights Reserved.

GPMS Local OD VisitORS Entry

## CREATE NEW ENTRY

Employee No. :

Employee Name :

Visit Location :

Purpose Of Visit :

Type Of Visit : Personal

Type of LocalOD : -- Select an option -- Official  Personal

Out Date and Time :

In Date and Time :

© 2023 - Gate-Pass Management . All Rights Reserved.

GPMS Local OD VisitORS Entry

## CREATE NEW ENTRY

Employee No. :

Employee Name :

Visit Location :

Purpose Of Visit :

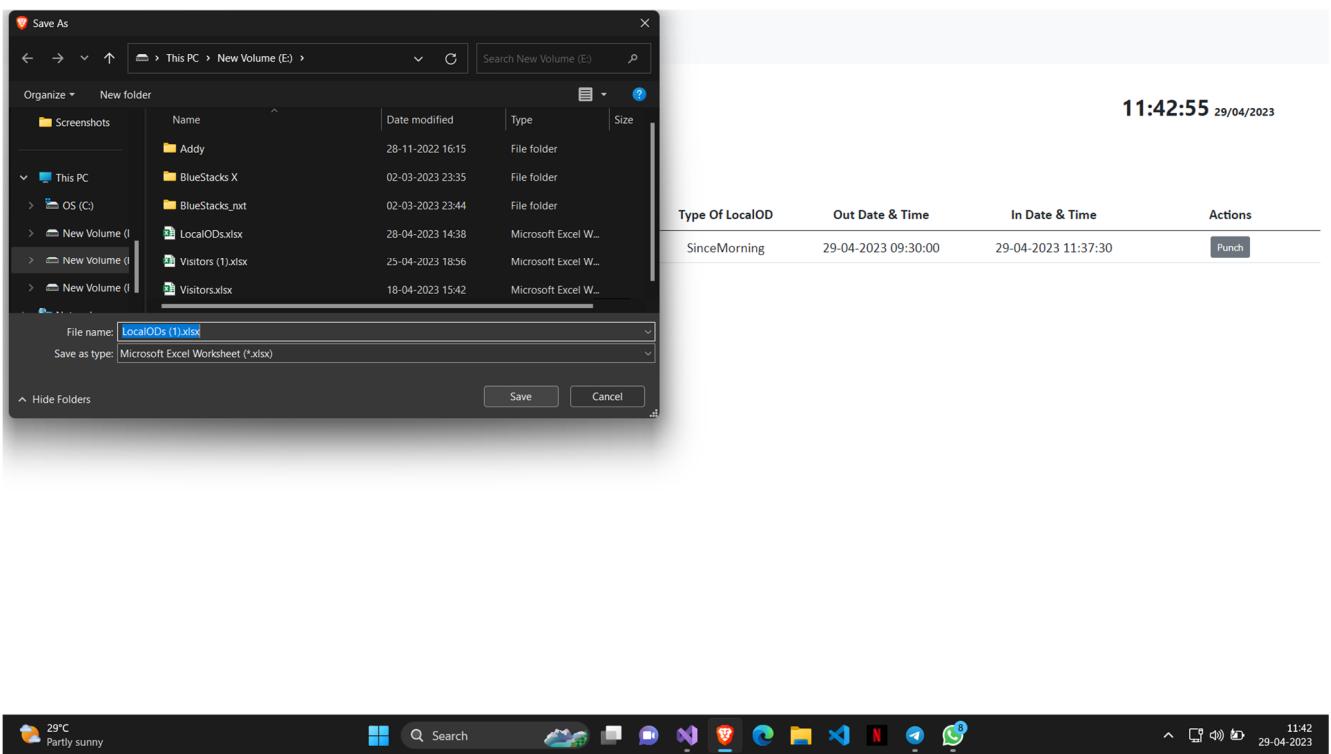
Type Of Visit : Personal

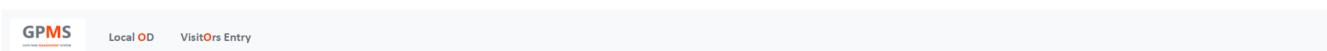
Type of LocalOD : -- Select an option --

Out Date and Time :

In Date and Time :

© 2023 - Gate-Pass Management . All Rights Reserved.





© 2023 - Gate-Pass Management . All Rights Reserved.

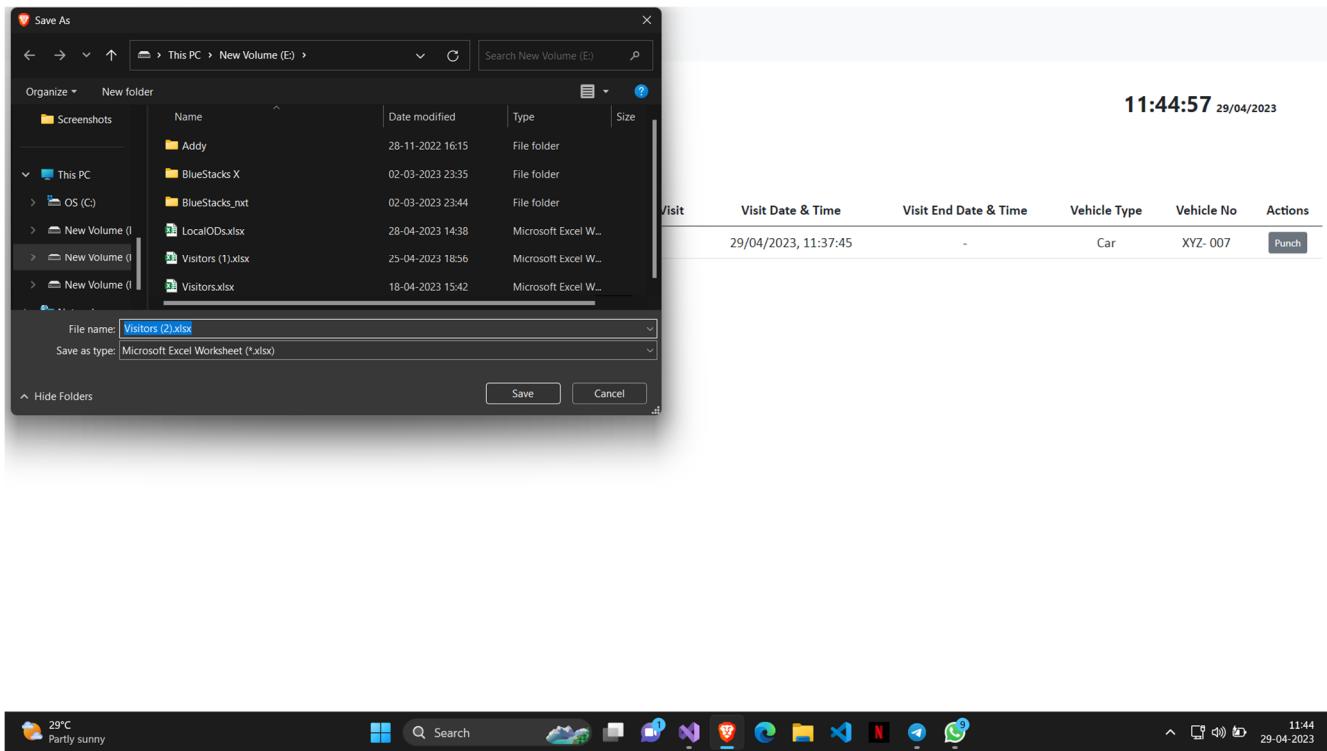
The screenshot shows the GPMS Create New Entry page. At the top, there are navigation links: 'GPMS' (with a logo), 'Local OD', and 'Visitors Entry'. Below this is a header with the text 'CREATE NEW ENTRY'.

The form consists of several input fields and dropdown menus:

- Entry Date and Time :**
- Visitors Mobile No. :**
- Visitor Name :**
- Company Name :**
- Employee Name :**
- Purpose Of Visit :**
- Visit Date and Time :**
- Visit End Date and Time :**
- Vehicle Type :**
- Vehicle No. :**

At the bottom right are two buttons: 'Create' (black button) and 'Cancel' (white button).

© 2023 - Gate-Pass Management . All Rights Reserved.



# CONCLUSION

In summary, the Gate Pass Management System is a useful application for managing and tracking the entry and exit of visitors and employees. The system includes several features, such as creating and managing gate passes, monitoring visitors' movement, generating reports, and more. The project is developed using ASP.NET MVC, Entity Framework, and SQL Server, which are popular and widely used technologies in web application development. The project is designed to be easy to use, and its user-friendly interface makes it accessible to anyone with basic computer knowledge. Overall, the Gate Pass Management System is an excellent tool for managing and tracking visitors and employees, and it can be easily customized to meet the specific needs of any organization.