**A PROJECT REPORT ON**

**“Student/Employee Management Portal”**

***Submitted in fulfillment of the***

***Requirement for the award of the degree***

**Of**

**2018-19**

**ACKNOWLEDGMENT**

Students/Employees are the backbone of any company therefore their management plays a major role in deciding the success of an organization.

Student/Employees Management System makes easy for employer to keep track of all records of employee. This software allows the administrator to edit employees, add new employees, and promote/terminate employees.

Each student/employee in the database is associated with a position can be added and edited when need arises.

The admin/employer can assign tasks to students/employees and assets their progress in order to keep track of student/employee performance.

It is simple to understand and can be used by anyone who is not even familiar with simple student/employees system.

INDEX

[1. Introduction 8](#_Toc450845206)

[1.1 Purpose of Document 8](#_Toc450845207)

[1.2 Scope of Project 8](#_Toc450845208)

[1.3 Objective 8](#_Toc450845209)

[1.4 Technology & literature review 8](#_Toc450845210)

[2. SYSTEM ANALYSIS 10](#_Toc450845211)

[2.1 Project Planning 10](#_Toc450845212)

[2.2 Risk Management 11](#_Toc450845213)

[2.3 General Description of project 12](#_Toc450845214)

[2.4 Functional Requirement 13](#_Toc450845215)

[2.5 Non-functional Requirement 13](#_Toc450845216)

[3. System Design 15](#_Toc450845217)

[3.1 Interface Design 15](#_Toc450845218)

[3.2 Use Case Diagram 16](#_Toc450845219)

[3.3 Class Diagram 18](#_Toc450845220)

[3.4 Activity Diagram 19](#_Toc450845221)

[3.5 Sequence Diagram 22](#_Toc450845222)

[4. Data Dictionary 25](#_Toc450845223)

[4.1 Table Name: ems\_employee 25](#_Toc450845224)

[4.2 Table Name: ems\_ task 26](#_Toc450845225)

[4.3 Table Name: ems\_ leave 27](#_Toc450845226)

[4.4 Table Name: ems\_ cities 28](#_Toc450845227)

[4.5 Table Name: ems\_ states 28](#_Toc450845228)

[4.6 Table Name: ems\_ countries 29](#_Toc450845229)

[4.7 Table Name: ems\_ event 29](#_Toc450845230)

[4.8 Table Name: ems\_ notes 30](#_Toc450845231)

[4.9 Table Name: ems\_ salary 31](#_Toc450845232)

[5. Implementation 32](#_Toc450845233)

[5.1 Description of developed system 32](#_Toc450845234)

[5.2 Security Feature 32](#_Toc450845235)

[5.3 Coding Standards 33](#_Toc450845236)

[6. Testing & verification 35](#_Toc450845237)

[6.1 Objectives of Software Testing 35](#_Toc450845238)

[6.2 Principles of Software Testing 35](#_Toc450845239)

[6.3 Testing Methods 36](#_Toc450845240)

[6.4 Strategy for Software Testing 37](#_Toc450845241)

[7. Screen Short 38](#_Toc450845242)

[7.1 Login Page 38](#_Toc450845243)

[8. Reference 49](#_Toc450845244)

Table of figure

[Figure 1 : Agile SDLC Model 11](file:///F:\onedrive\sid-JaiswalSiddhart.docx#_Toc450837215)

[Figure 2 : Use Case Diagram for Employee 16](file:///F:\onedrive\sid-JaiswalSiddhart.docx#_Toc450837216)

[Figure 3 : Use Case Diagram for Admin 17](file:///F:\onedrive\sid-JaiswalSiddhart.docx#_Toc450837217)

[Figure 4 : Class Diagram 18](file:///F:\onedrive\sid-JaiswalSiddhart.docx#_Toc450837218)

[Figure 5 : Activity Diagram for Leave Application Submission 19](file:///F:\onedrive\sid-JaiswalSiddhart.docx#_Toc450837219)

[Figure 6 : Activity Diagram for Leave Approve/Reject 20](file:///F:\onedrive\sid-JaiswalSiddhart.docx#_Toc450837220)

[Figure 7 : Task Assign to Employee 21](file:///F:\onedrive\sid-JaiswalSiddhart.docx#_Toc450837221)

[Figure 8 : Sequence Diagram for Add New Employee 22](#_Toc450837222)

[Figure 9 : Sequence Diagram for Create Task 23](#_Toc450837223)

[Figure 10 : Sequence Diagram for Create Leave Application 23](#_Toc450837224)

[Figure 11: Sequence Diagram for Approve/Reject Leave Application 24](#_Toc450837225)

# Introduction

## Purpose of Document

This document specifies the complete report for the Student/Employee Management System (MANAGIO). It describes scope of the system, both functional and non-functional requirements for the software, design constraints and system interfaces, user manual, etc.

## Scope of Project

The Scope of this project is expected to be user friendly and will offer easy access to data as well as services such as monitoring employee, task management, online electronic leave management and employee tracking.

The employee is expected to have direct interaction with this system through a password protected user account therefore proposed system is web based to enable accessibility

## Objective

In this world of growing technologies everything has been computerized with large number of work opportunities the Human workforce has increased.

Thus there is a need of a system which can handle the data of such a large number of Employees. This project simplifies the task of maintaining records because of its user friendly nature.

The objective of this project is to provide a comprehensive approach towards the management of employee information. This will be done by designing and implementing an Employee Management System that will bring up a major paradigm shift in the way that employee information is handled.

## Technology & literature review

|  |  |
| --- | --- |
| Platform | Web Application and Android |
| Front End | HTML5, CSS3, Java Script, Android |
| Back End | PHP5, MySQL |
| Developing Tool | Adobe Dreamweaver CS5, Android Studio |
| Presentation Tool | MS-PowerPoint, MS-Word |

* **HTML**
* To publish information for global distribution, one needs a universally understood language, a kind of publishing mother tongue that all computers may potentially understand. The publishing language used by the World Wide Web is HTML (from Hypertext Mark-up Language).
* HTML gives authors the means to publish online documents with headings, text, tables, lists, photos, etc.
* Retrieve online information via hypertext links, at the click of a button.
* Design forms for conducting transactions with remote services, for use in searching for information, making reservations, ordering products, etc.
* Include spreadsheets, video clips, sound clips, and other applications directly in their documents.
* **CSS**
* CSS stands for Cascading Style Sheets. Styles sheets define HOW HTML elements are to be display, just as the font tag and the color attribute in HTML 3.2. Styles are normally save in external CSS files. External style sheets enable you to change the appearance and layout of all the pages in your Web, just by editing one single CSS document.
* CSS is a breakthrough in Web design because it allows developers to control the style and layout of multiple Web pages all at once. As a Web developer, you can define a style for each HTML element and apply it to as many Web pages as you want. To make a global change, simply change the style, and all elements in the Web are update automatically.
* Multiple Styles Will Cascade Into One.
* Style sheets allow style information to be specify in many ways. Styles can be specify inside a single HTML element, inside the <head> element of an HTML page, or in an external CSS file. Even multiple external style sheets can be reference inside a single HTML document.
* **PHP**
* PHP is an HTML-embedded scripting language. Much of its syntax is borrow from C, Java and Perl with a couple of unique PHP-specific features thrown in. The goal of the language is to allow web developers to write dynamically generated pages quickly.
* PHP stands for PHP: Hypertext Pre-processor. This confuses many people because the first word of the acronym is the acronym. This type of acronym is call a recursive acronym.
* **JavaScript**
* JavaScript can put dynamic text into an HTML page - A JavaScript statement like this: document. Write ("<h1>" + name + "</h1>") can write a variable text into an HTML page.
* JavaScript can react to events - A JavaScript can be set to execute when something happens, like when a page has finished loading or when a user clicks on an HTML element.
* JavaScript can read and write HTML elements - A JavaScript can read and change the content of an HTML element.
* JavaScript is use to give validation data - A JavaScript can be used to validate form data before it is submitted to a server, this will save the server from extra processing.

1. **SYSTEM ANALYSIS**

System analysis is done by developers to decide a roadmap of their plan and try to bring up the best software model suitable for the project.

System analysis includes Understanding of software product limitations, learning system related problems or changes to be done in existing systems beforehand, identifying and addressing the impact of project on organization and personnel etc.

The project team analyse the scope of the project and plans the schedule and resources accordingly.

* 1. **Project Planning**

Planning is to deciding what type of activities we will engage in our Project.

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product.

Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing.

* **Individuals and interactions:**

In agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.

* **Working software:**

Demo working software is considered the best means of communication with the customer to understand their requirement, instead of just depending on documentation.

* **Customer collaboration:**

As the requirements cannot be gathered completely in the beginning of the project due to various factors, continuous customer interaction is very important to get proper product requirements.

* **Responding to change:**

Agile development is focused on quick responses to change and continuous development.

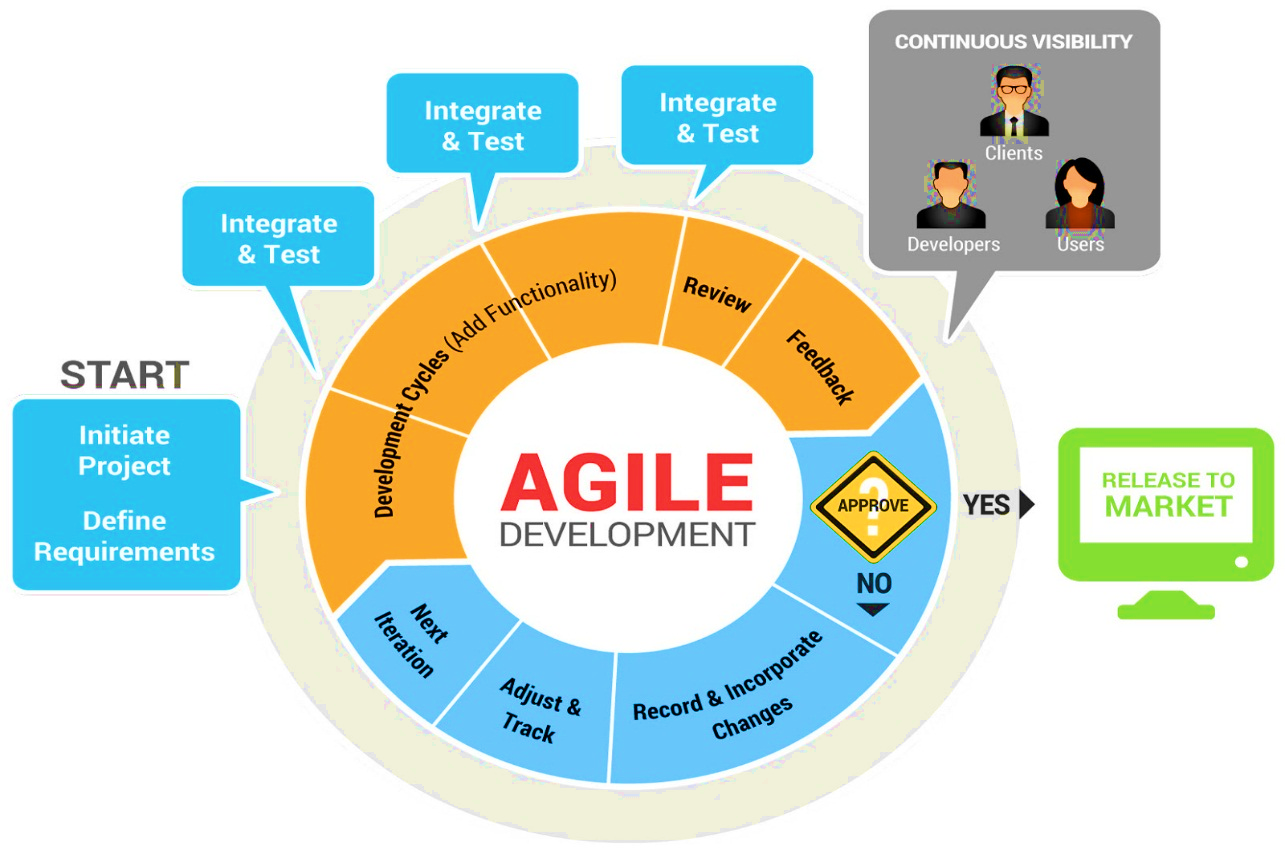


Figure 1 : Agile SDLC Model

* 1. **Risk Management**

When some disturbance occur & that is risky for our Project but we find the path & solve the risk that’s call risk management.

* **Financial Risk:**

The Degree of uncertainly is that project budget will be maintain. For example: - The project consumes more cost (finance) then estimated.

* **Technical Risk:**

If Technical Risk becomes implementation may become difficult or impossible. For example: - The database used in the system cannot run as many transactions.

* **Project Risk:**

If Project Risk occurs it will slip the project and because of this cost will increase. For example: - The project crash when the project run.

* **Detectable Risk:**

It is the application from past project experience. For example: - The project is already mead in present. **“It can be the crime of copy case.”**

* **Organization Risk:**

Organization says do not make project because organization breakdown or some one other make the project in less time or cost. For example: - May the organization reconstruct the same project from other?

* **Estimation Risk:**

The time required to develop the software is underestimated. For example: - The codding consumes more time then we decided before in planning

* **Business Risks:**

Threaten the viability of the s/w to be built. For example: - the project is threatened by Business Partner.

* **Known Risks:**

Those can be uncovered after carefully evaluating project plan. For example: - the project is going wrong because of spelling mistakes.

* **Detectable Risks:**

Detectable Risks are extrapolated from past project experience.

* 1. **General Description of project**

The general description will be limited to the following:

* **Employee profiles:**

Employees will have access to their personal profiles and will be able to edit their details.

* **Electronic leave application:**

Complete elimination of paperwork in leave management by enabling an employee apply for leave as well as check their leave status through the system. This will also enable the admin to accept/reject leave application through the system.

* **Task Management:**

Assign tasks and projects to employees, assign a project team and keep track of the progress.

* **Salary Management:**

System should be able to generate the following salary information:

* Salary calculation for every employee for per month and per annum.
* Total salary calculation for each month for the whole company.
* Total salary calculation for a year for the whole company.
* Total salary of the organization per month and per annum.
* Total salary calculation for every domain of the company, for every month and for the whole year.
  1. **Functional Requirement**

The Function of this project are as following:

* **Authentication:**
* **Login: -** The user can login to the Employee Management System with his/her username and password.
* **Logout: -** The user can log out from the Employee Management System.
* **Login failure: -** If the user does not exist in the database or the user has not authorized by the admin.
* **Authorization:**
* User role check- After logging in, the user role will be check from the database and the user interface will be display according to their role.
* **Process Data:**
* **Display: -** User with defined roles can display the content of the database. Being more specific, employee can only view his/her personal information. Admin cannot only see his/her personal information but also employee’s information who are under his/her department. Admin can display their personal information and all employees’ information.
* **Edit: -** A user employee can edit his/her specific personal information. Admin can edit all information related to all employees’ including their user role type.
* **Leave Application/Approval:**
* **Leave application: -** The user can be able to fill in leave application form in the appropriate fields.
* **Leave approval: -** The admin can be able to approve leave applications based on the reasons stated, length of leave
* **Leave days accrued: -** The user shall be able to check the number of leave days accrued.
* **Task Management:**
* **Create Task team: -** The Admin is able to create a project and come up with a project team.
* **Work Breakdown structure: -** The Admin shall be able to assign tasks to the task team as well as monitor their progress.

* 1. **Non-functional Requirement**
* **Hardware requirements**

Employee Management System should be able to work on a computer with the following minimum hardware specifications:

CPU: Pentium III (500MHz) and above

Memory: 128 MB and above

Capacity: 4GB of hard drive to server side

* **Software requirements**

Employee Management System application is a web-based application, internet connection must be established.

The Employee Management System software personal database model will support MySQLi environment as DBMS.

We can use intranet to make website secure to make access only from company campus.

1. **System Design**

System Design is to bring down whole knowledge of requirements and analysis on the desk and design the software product. The inputs from users and information gathered are the inputs.

* 1. **Interface Design**

The web application was created with the following design considerations in mind:

* **Consistent: -** The website should have a similar look and feel on every page. Every page should have the same header/logo, heading style, fonts, navigations etc.
* **Efficient and easy to maintain: -** This refers to the fact that there is need to separate content from layout, so that you can easily change your page design without editing every page on the site.
* **Layout**: - The layout of each page should have a good contrast between the text and background area. This helps considerably with visibility as it will be difficult to read the text if it is almost the same color as the background. Monitor size should also be taken into consideration.
* **Easy to navigate and use: -** Users should not have a hard time trying to navigate the site. Navigation links should be consistent and clearly labeled. All navigation links should also be working properly and should point to the intended page/site.
* **Browser compatible: -** When designing the site consider different browser environments. Extensive testing should be done on each page in all the major browsers and the design changed appropriately to cater for all.
* **Visually appealing: -** The use of color, text, fonts and graphics should be carefully considered and used to ensure that the site is visually appealing to its visitors.
* **Speed: -** The performance of a website is mostly rated by its up -time and downtime. These terms refers to the amount of time it takes the site to respond to requests. Graphics should be kept to a minimum to allow the site to load faster. The pages on the site should load within an acceptable time
* **Database Table Design: -** The database must have well designed for maintenance. Table name must start with database name then ‘\_’ (underscore). For Example: - “ems\_employee”.
* **Database Table column Design: -** The database table column name start with small letter then space is replace by second letter capital. For Example: - “startDate”.
* **Coding: -** Coding must be well formatted by giving right tab spacing & well commented code.
  1. **Use Case Diagram**

A **use case diagram** at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different **use** cases in which the user is involved.

* + 1. **Use Case Diagram for Employee**

Figure 2 : Use Case Diagram for Employee

* + 1. **Use Case Diagram for Admin**

Figure 3 : Use Case Diagram for Admin

* 1. **Class Diagram**

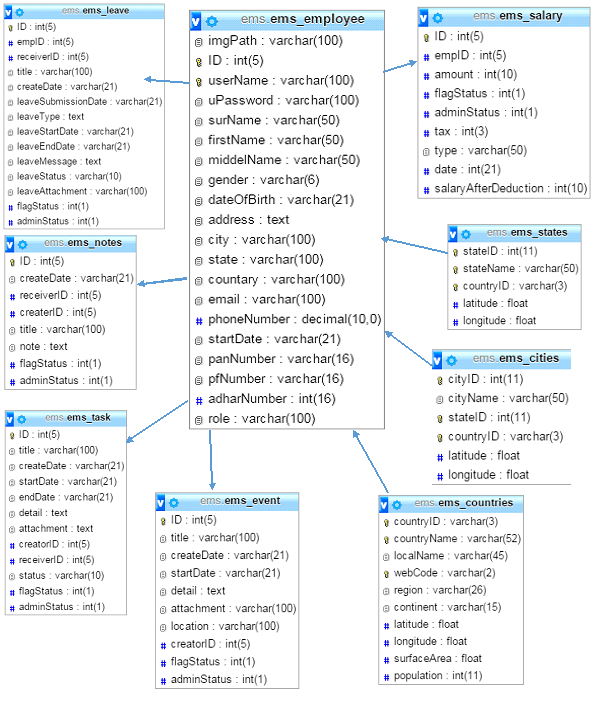
****a **class diagram** in the Unified Modelling Language (UML) is a type of static structure **diagram** that describes the structure of a system by showing the system's **classes**, their attributes, operations (or methods), and the relationships among objects.

Figure 4 : Class Diagram

* 1. **Activity Diagram**

**Activity diagram** is another important **diagram** in UML to describe dynamic aspects of the system. **Activity diagram** is basically a flow chart to represent the flow form one **activity** to another **activity**. The **activity** can be described as an operation of the system. So the control flow is drawn from one operation to another.

* + 1. **Leave Application Submission**

Figure 5 : Activity Diagram for Leave Application Submission

* + 1. **Leave Approve/Reject**

Figure 6 : Activity Diagram for Leave Approve/Reject

* + 1. **Task Assign to Employee**

Figure 7 : Task Assign to Employee

* 1. **Sequence Diagram**

A **Sequence diagram** is an interaction **diagram** that shows how processes operate with one another and in what order. It is a construct of a Message **Sequence** Chart. A **sequence diagram** shows object interactions arranged in time **sequence**.

* + 1. **Add New Employee**

Figure 8 : Sequence Diagram for Add New Employee

* + 1. **Sequence Diagram for Create Task**



Figure 9 : Sequence Diagram for Create Task

* + 1. Create Leave Application



Figure 10 : Sequence Diagram for Create Leave Application

* + 1. Approve/Reject Leave Application



Figure 11: Sequence Diagram for Approve/Reject Leave Application

1. **Data Dictionary**

A structured place to keep details of the contents of data flows, processes, and data store. It is a structured repository of data about data. It is a set of definitions of all DFD elements.

Items to be defined in Data Dictionary:

* **Data Elements: -**

Smallest unit of data that provides for no further decomposition. For example: date consists of day, month and year

* **Data Structure: -**

A group of data elements handled as a unit. For example: phone is a data structure consisting of four data elements: area-code-exchange-number-extension.

* **Data Flows and Data Stores: -**

Data flows are data structures in motion, whereas data stores are data structures at rest. A data store is a location where data structures are temporarily located.

## Table Name: ems\_employee

**Table Description: This Table is contain All Information of Employee.**

**Unique Key: ID, userName**

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Null** |
| imgPath | varchar(100) | No |
| ID | int(5) | No |
| userName | varchar(100) | No |
| uPassword | varchar(100) | No |
| surName | varchar(50) | No |
| firstName | varchar(50) | No |
| middelName | varchar(50) | No |
| Gender | varchar(6) | No |
| dateOfBirth | varchar(21) | No |
| Address | text | No |
| City | varchar(100) | No |
| State | varchar(100) | No |
| Countary | varchar(100) | No |
| Email | varchar(100) | No |
| phoneNumber | decimal(10,0) | No |
| startDate | varchar(21) | No |
| panNumber | varchar(16) | No |
| pfNumber | varchar(16) | No |
| adharNumber | int(16) | No |
| Role | varchar(100) | No |

## Table Name: ems\_ task

**Table Description: This Table is contain All Information of Task.**

**Unique Key: ID**

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Null** |
| ID | int(5) | No |
| Title | varchar(100) | No |
| createDate | varchar(21) | No |
| startDate | varchar(21) | No |
| endDate | varchar(21) | No |
| Detail | text | No |
| attachment | text | No |
| creatorID | int(5) | No |
| receiverID | int(5) | No |
| Status | varchar(10) | No |
| flagStatus | int(1) | No |
| adminStatus | int(1) | No |

## Table Name: ems\_ leave

**Table Description: This Table is contain All Information of Leave Application.**

**Unique Key: ID**

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Null** |
| ID | int(5) | No |
| empID | int(5) | No |
| receiverID | int(5) | No |
| Title | varchar(100) | No |
| createDate | varchar(21) | No |
| leaveSubmissionDate | varchar(21) | No |
| leaveType | text | No |
| leaveStartDate | varchar(21) | No |
| leaveEndDate | varchar(21) | No |
| leaveMessage | text | No |
| leaveStatus | varchar(10) | No |
| leaveAttachment | varchar(100) | No |
| flagStatus | int(1) | No |
| adminStatus | int(1) | No |

## Table Name: ems\_ cities

**Table Description: This Table is contain List of all cities.**

**Unique Key: cityID**

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Null** |
| cityID | int(11) | No |
| cityName | varchar(50) | No |
| stateID | int(11) | No |
| countryID | varchar(3) | No |
| latitude | Float | No |
| longitude | Float | No |

## Table Name: ems\_ states

**Table Description: This Table is contain List of all states.**

**Unique Key: stateID**

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Null** |
| stateID | int(11) | No |
| stateName | varchar(50) | No |
| countryID | varchar(3) | No |
| latitude | float | No |
| longitude | float | No |

## Table Name: ems\_ countries

**Table Description: This Table is contain List of all countries.**

**Unique Key: countryID**

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Null** |
| countryID | varchar(3) | No |
| countryName | varchar(52) | No |
| localName | varchar(45) | No |
| webCode | varchar(2) | No |
| Region | varchar(26) | No |
| continent | varchar(15) | No |
| Latitude | float | No |
| longitude | float | No |
| surfaceArea | float | No |
| population | int(11) | No |

## Table Name: ems\_ event

**Table Description: This Table is contain all Information about event.**

**Unique Key: ID**

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Null** |
| ID | int(5) | No |
| Title | varchar(100) | No |
| createDate | varchar(21) | No |
| startDate | varchar(21) | No |
| Detail | text | No |
| attachment | varchar(100) | No |
| location | varchar(100) | No |
| creatorID | int(5) | No |
| flagStatus | int(1) | No |
| adminStatus | int(1) | No |

## Table Name: ems\_ notes

**Table Description: This Table is contain List of all countries.**

**Unique Key: ID**

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Null** |
| ID | int(5) | No |
| createDate | varchar(21) | No |
| receiverID | int(5) | No |
| createrID | int(5) | No |
| Title | varchar(100) | No |
| Note | text | No |
| flagStatus | int(1) | No |
| adminStatus | int(1) | No |

## Table Name: ems\_ salary

**Table Description: This Table is contain all Information of salary.**

**Unique Key: stateID**

|  |  |  |
| --- | --- | --- |
| **Column** | **Type** | **Null** |
| ID | int(5) | No |
| empID | int(5) | No |
| Amount | int(10) | No |
| flagStatus | int(1) | No |
| adminStatus | int(1) | No |
| Tax | int(3) | No |
| Type | varchar(50) | No |
| Date | int(21) | No |
| salaryAfterDeduction | int(10) | No |

1. **Implementation**

The developed system encompasses various activities associated with managing employee information.

The main functionalities available in this system are:

* Maintaining Employee Profiles
* Task Management
* Leave Management
* Note Management

All these features include the ability to add user, update (edit), and retrieve through search results. It also contains a report generation system that can be saved in a pdf file format.

The system works in following manner:

* 1. **Description of developed system**

All users are presented with the same login interface. User must login the system by means of valid username/password combination.

After access is granted to the system, the admin can add a new user to the system by entering the basic information which are the full names and email address.

The admin also assigns the new user a role which will determine the access level. During the process of user registration, the all users are issued with a unique username and password combination.

Admin can assign Task to Employee and Employee can see his task.

Employees can apply for leave by filling in a form as well as submitting an attachment to support their leave request.

Admin can Approve or Reject Leave application

Upon logging in to the system, the Admin gets notifications on the task update, leave applications, notes etc.

Upon logging in to the system, the Employee also gets notifications on the task update, leave applications, notes etc.

* 1. **Security Feature**

PHP is a very potential technology for developing website. Now a day many applications are being developed in PHP. For any website Security is a very important feature. Sometimes extra security features provided in the website may degrade the overall performance of the website. So one should properly choose security features so as the application is reliable as well as performance is also not very much affected. This article considers the security features in PHP But these are applicable to higher versions also.

We can divide the whole Security feature analysis to following major categories:

* Authentication
* Configuration
* Data Access Security
* Code Access Security
* Exception Handling
* Communication Security
  1. **Coding Standards**
* **Naming Guidelines/convention:**
* **Web page name:**

Rule: - The Web Page Name Must Be in small letter, Space is replace by “-”.

Example: - db-create-employee.php

* **Table:**

Rules: - First Table name contain database Name then “\_” then table name

Examples: - ems\_employee, ems\_leave

* **Database Field Name:**

Rule: - The Field Name Must Be in small letter, Space is replacing by Capitalizing next character after space.

Examples: - firstName, lastName

* **General Rules:** 
  + Do not use spaces is the naming of Database object.
  + Do not use SQL keyword as the name of Database object.
  + In the eases where this is necessary, surround the object name with brackets, such as [‗year‘].
  + Do not prefix stored procedure with ‗sp\_2‘.
  + Prefix table name with the owner name3.
  + For example : - leave, password, join, order
* **Coding Guidelines:**
  + Optimize queries using tool provide by SQL server.
  + Access tables the same order in your stored procedure and triggering consistently.
  + We do not call function repeatedly within us stored procedure, trigger, function and batches.
  + Avoid wild-card character at the beginning of the word while searching using the LIKE keyword, as this result in an index scan, which purpose of an index.
  + Do not use white space in identifiers.
* **Formatting Guidelines:** 
  + Use upper case for all SQL keywords- SELECT, INSERT and UPDAT.
  + Indent code to improve readability.
  + Cement code blocks that are not easy under stable- use single line comment marks (--), reserve multi-comments (/\*…….\*/) for multi blocking out selection of code.
  + Use parenthesis to increase readability-WHERE (productName='$searchKeyword').
  + Use one blank line to separate code section.
  + Use space so that expression read like sentence – fill factor = 25, not fill factor=25.

1. **Testing & verification**

Software testing is the process of executing a program with intension of finding errors in the code. It is a process of evolution of system or its parts by manual or automatic means to verify that it is satisfying specified or requirements or not. Generally, no system is perfect due to communication problems between user and developer, time constraints, or conceptual Mistakes by developer. To purpose of system testing is to check and find out these errors or faults as early as possible so losses due to it can be saved.

Testing is the fundamental process of software success.

Testing is not a distinct phase in system development life cycle but should be applicable throughout all phases i.e. design development and maintenance phase.

Testing is used to show incorrectness and considered to success when an error is detected.

* 1. **Objectives of Software Testing**

The software testing is usually performed for the following objectives:-

* **Software Quality Improvement: -**

The computer and the software are mainly used for complex and critical applications and a bug or fault in software causes severe losses. So a great consideration is required for checking for quality of software.

* **Verification and Validation:-**

Verification means to test that we are building the product in right way .i.e. are we using the correct procedure for the development of software so that it can meet the user requirements.

Validation means to check whether we are building the right product or not.

* **Software Reliability Estimation:** -

The objective is to discover the residual designing errors before delivery to the customer. The failure data during process are taken down in order to estimate the software reliability.

* 1. **Principles of Software Testing**

Software testing is an extremely creative and challenging task. Some important principles of software testing are as given:-

**Principle 1: Definition** (To test a program is to try to make it fail.)

This keeps the testing process focused: Its single goal is to uncover faults by triggering failures. Any inference about quality is the responsibility of quality assurance but beyond the scope of testing. The definition also reminds us that testing, unlike debugging, does not deal with correcting faults, only finding them.

**Principle 2: Tests versus specs** (Tests are no substitute for specifications.)

The danger of believing that a test suite can serve as specification is evidenced by several software disasters that happened because no one had thought of some extreme case. Although specifications can miss cases too, at least they imply an effort at generalization.

**Principle 3: Regression testing** (Any failed execution must yield a test case, to remain a permanent part of the project‘s test suite.)

This principle covers all failures occurring during development and testing. It suggests tools for turning a failed execution into a reproducible test case, as have recently emerged.

**Principle 4:** (variant)

Determining test success or failure should be an automatic process consisting of monitoring contract satisfaction during execution. The principle subsumes the previous one but is presented as a variant so that people who do not use contracts can retain the weaker form.

**Principle 5: Manual and automatic test cases** (An effective testing process must include both manually and automatically produced test cases.)

Manual tests are good at depth: They reflect developers ‘understanding of the problem domain and data structure. Automatic tests are good at breadth: They try many values, including extremes that humans might miss.

**Principle 6: Empirical assessment**

Empirical assessment of testing strategies Evaluate any testing strategy, however attractive in principle, through objective assessment using explicit criteria in a reproducible testing process.

**Principle 7: Assessment criteria**

Assessment criteria A testing strategy‘s most important property is the number of faults it uncovers as a function of time.

* 1. **Testing Methods**

Software testing methods are traditionally divided into white- and black-box testing.

* **White box testing : -**

White box testing is when the tester has access to the internal data structures and algorithms including the code that implement these. The following types of white box testing exist:

* **Code coverage : -**

Creating tests to satisfy some criteria of code coverage (e.g., the test designer can create tests to cause all statements in the program to be executed at least once).

* **Fault injection methods : -**

Improving the coverage of a test by introducing faults to test code paths.

* **Static testing : -**

White box testing includes all static testing.

* **Black box Testing:**

Black box testing treats the software as a "black box"—without any knowledge of internal implementation. Black box testing methods include:

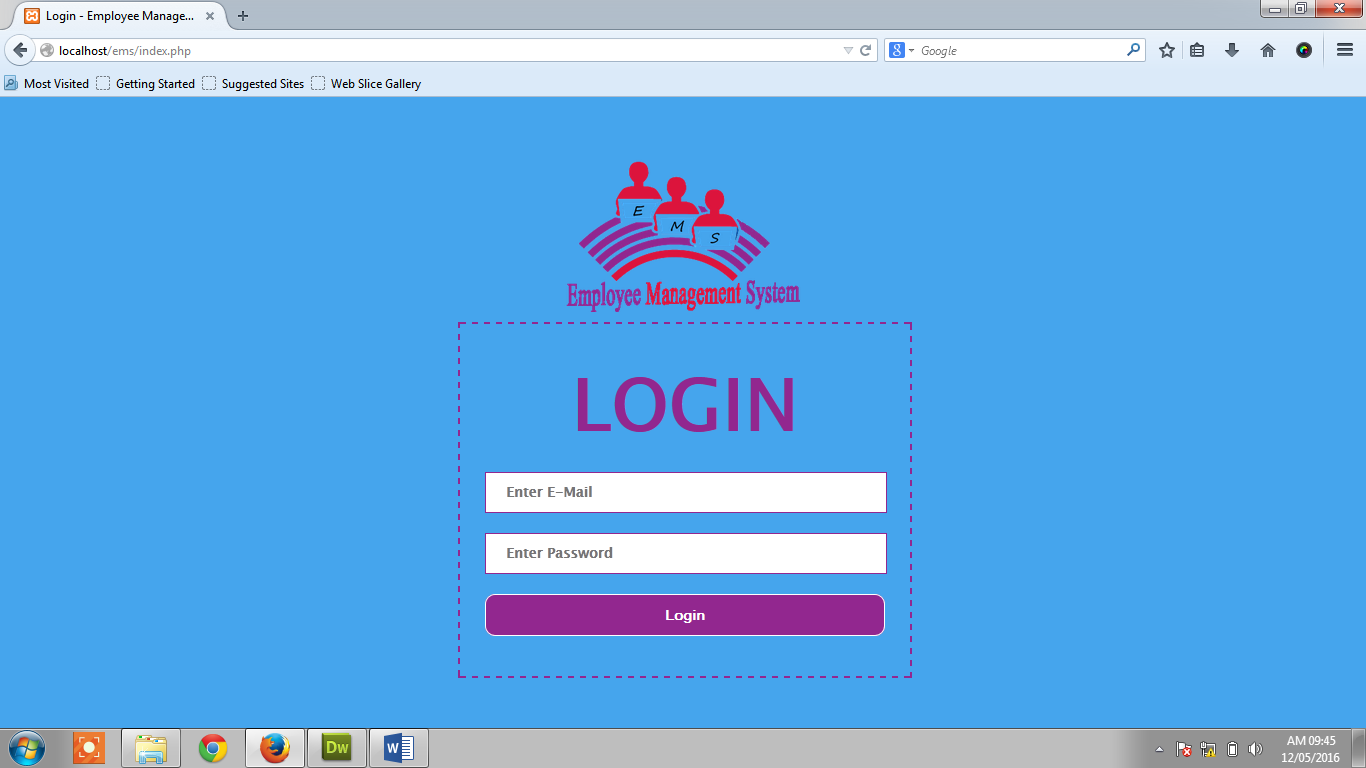
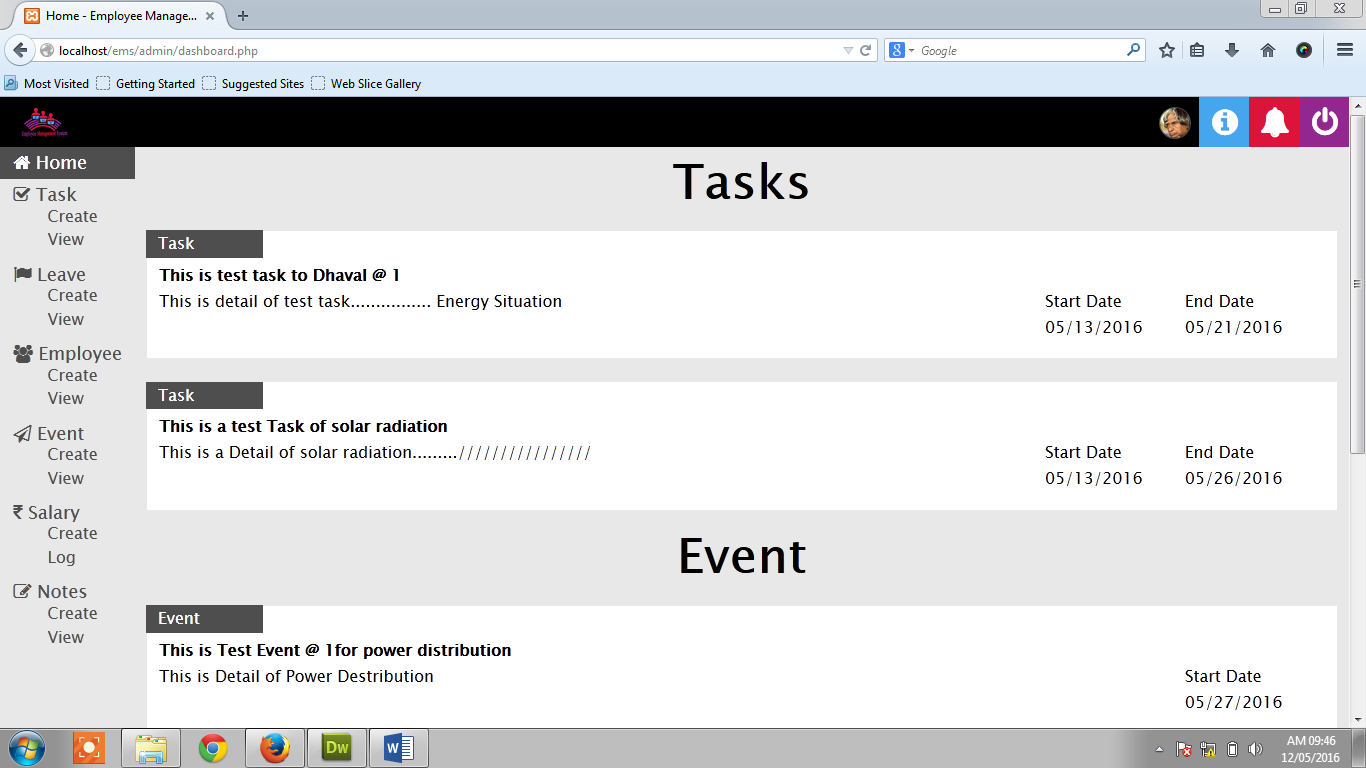
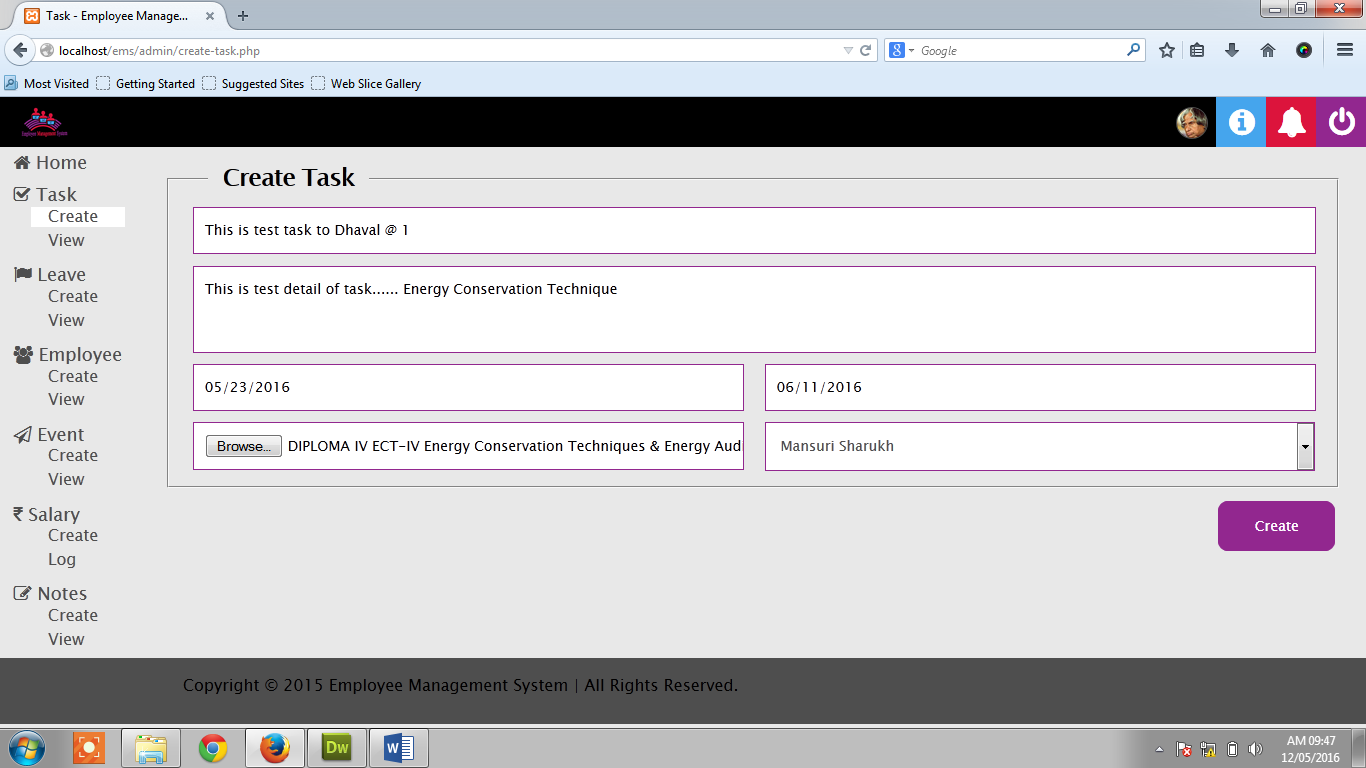
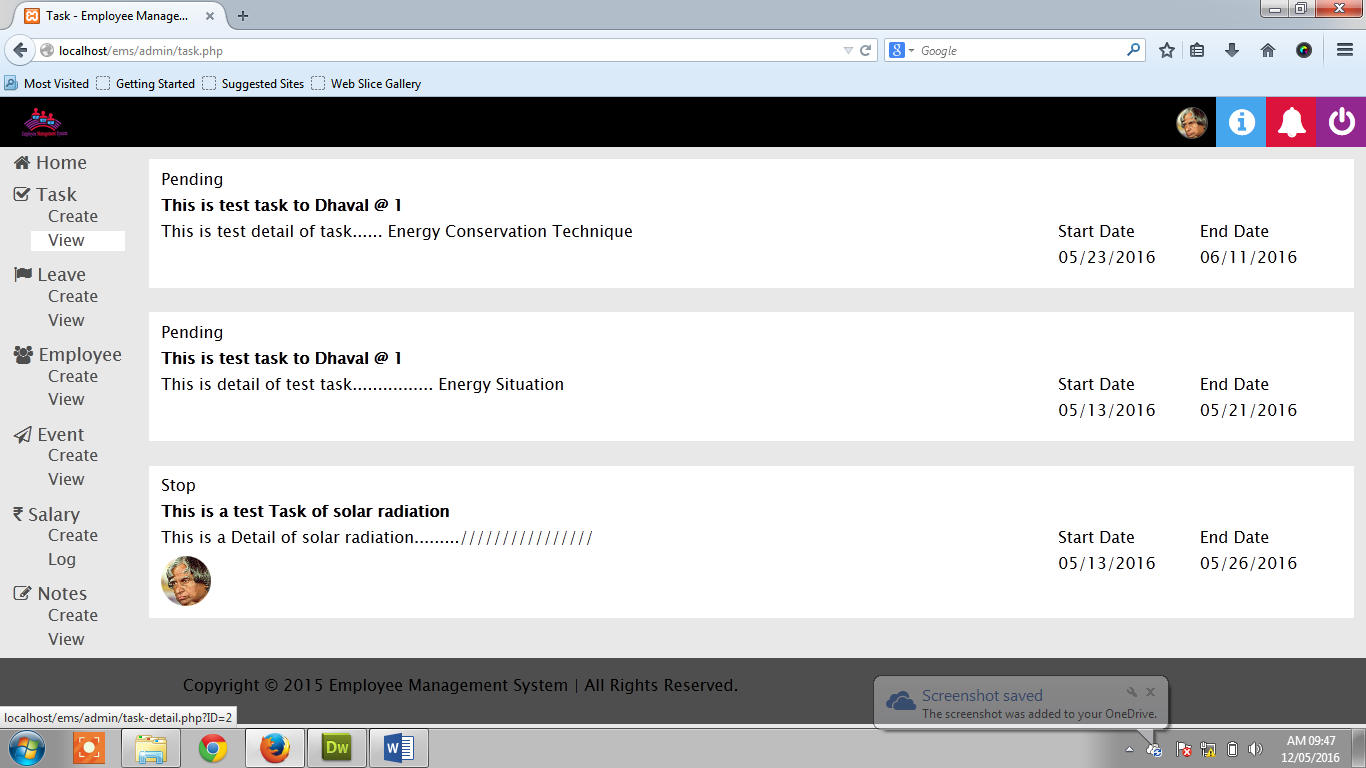
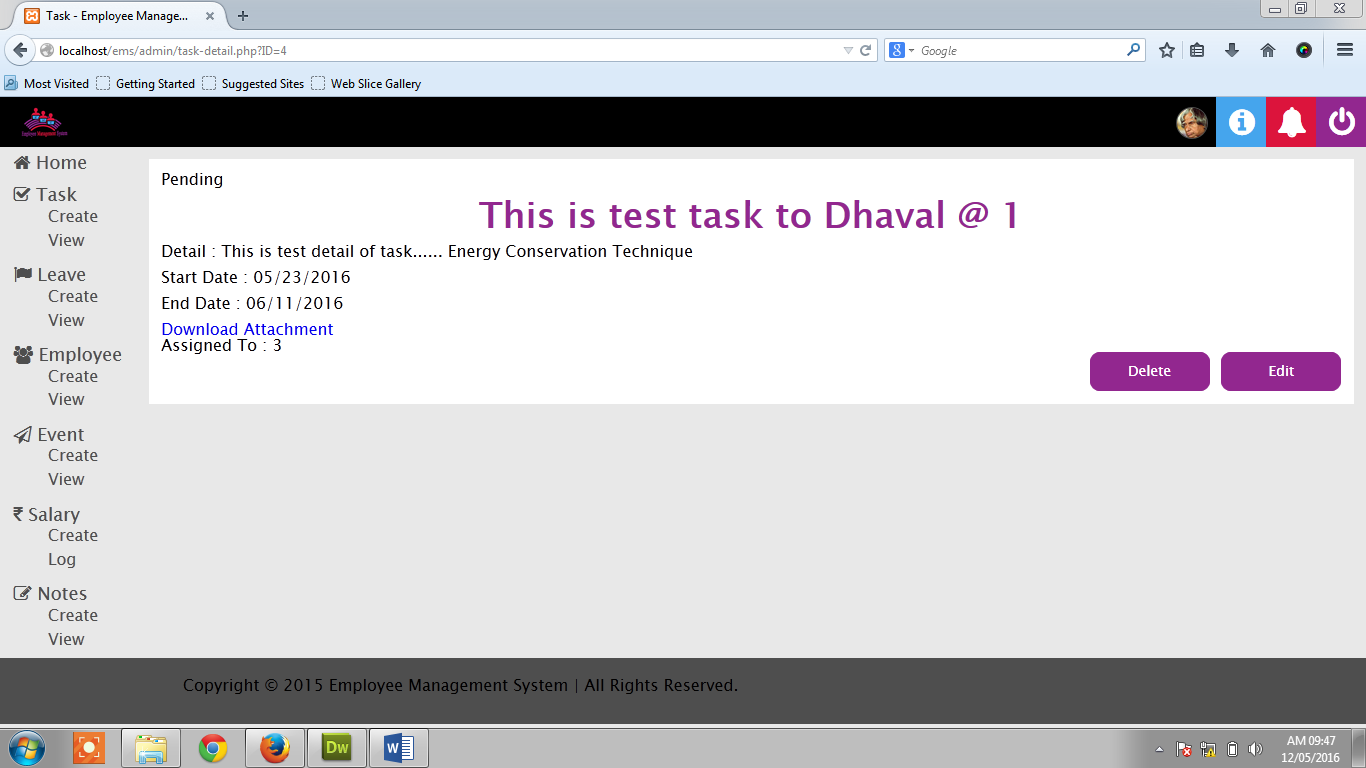
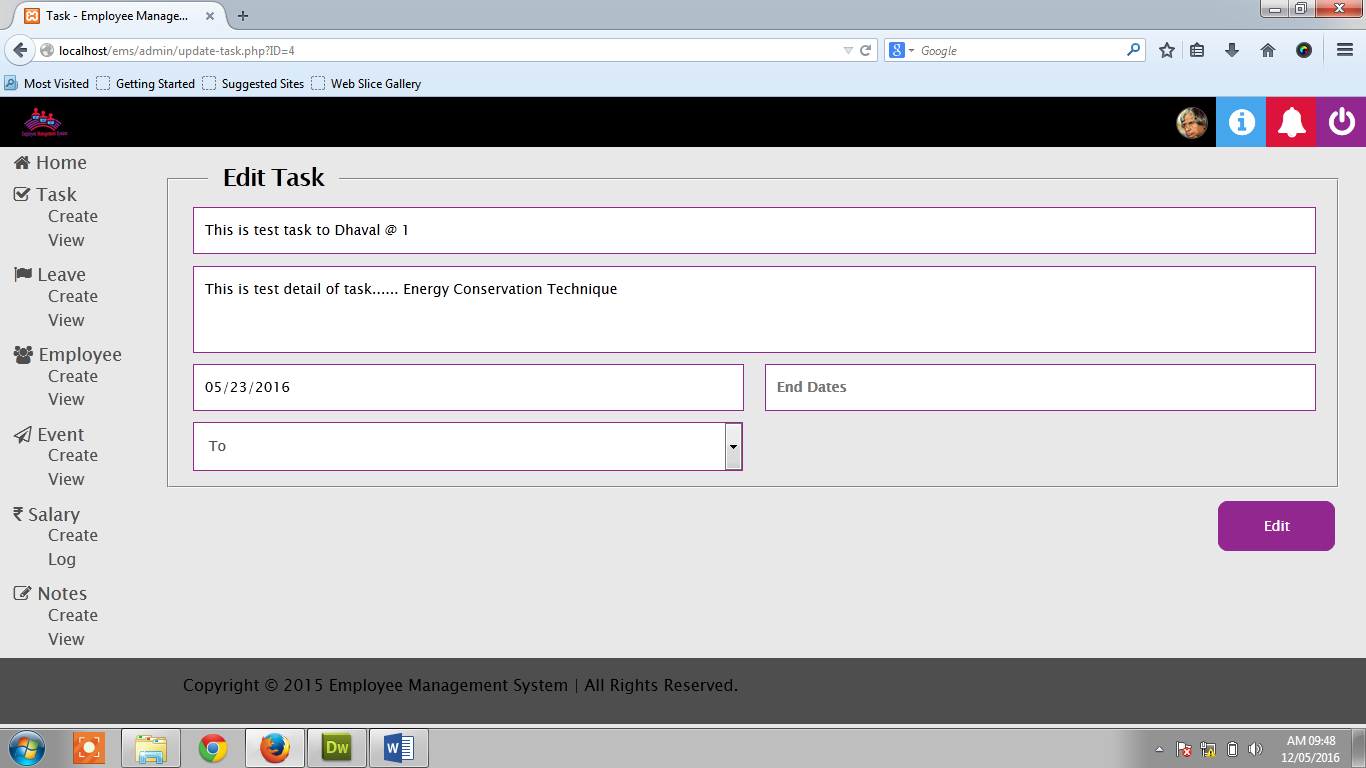
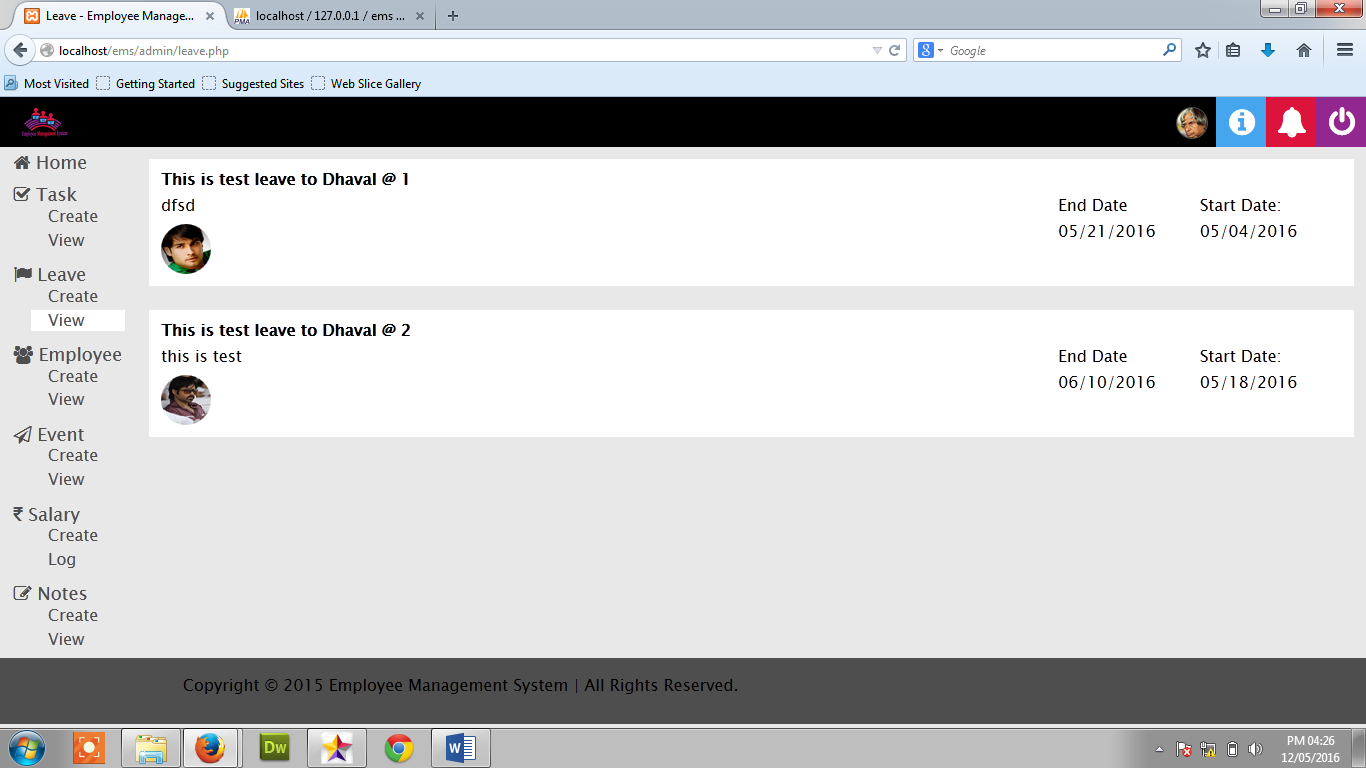
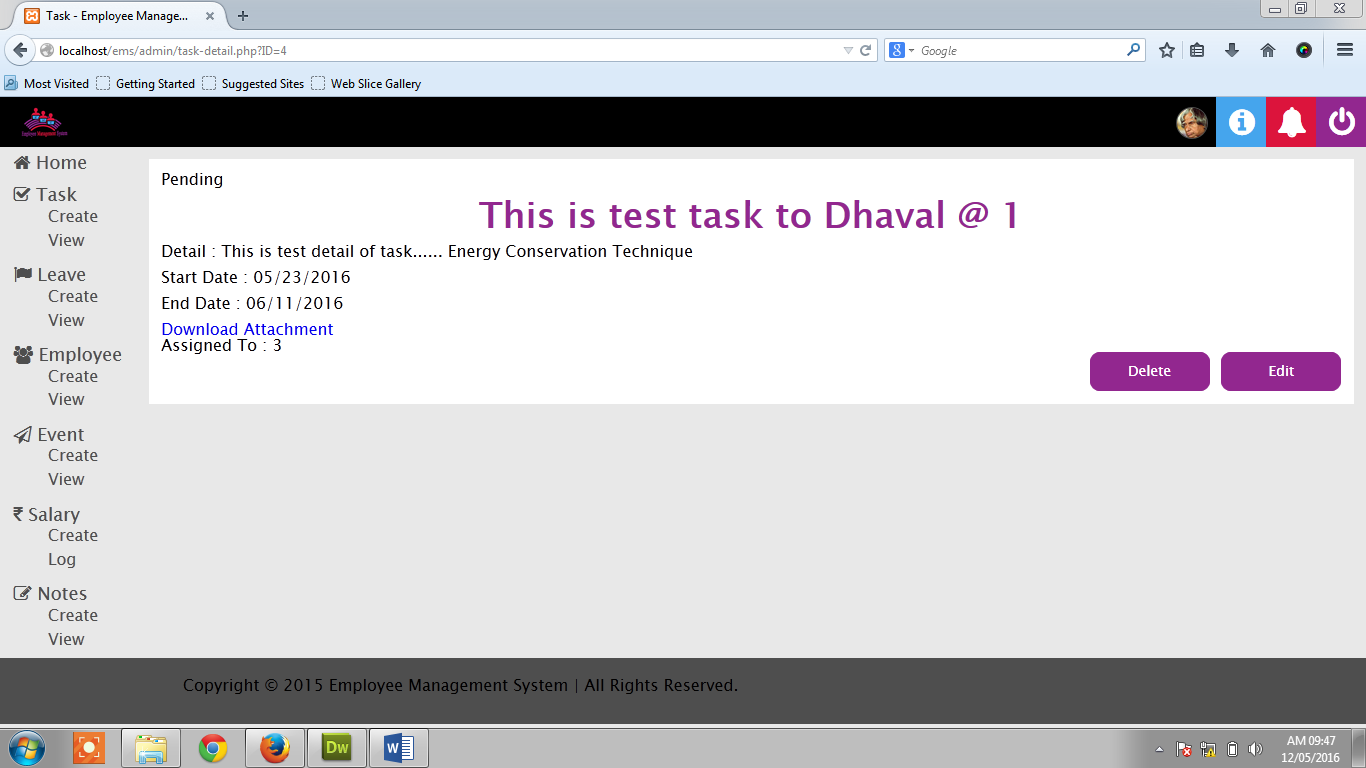
* Equivalence partitioning
* Boundary value analysis
* All-pairs testing
* Fuzz testing
* Model-based testing
* Exploratory testing
* Specification-based testing.
  1. **Strategy for Software Testing**

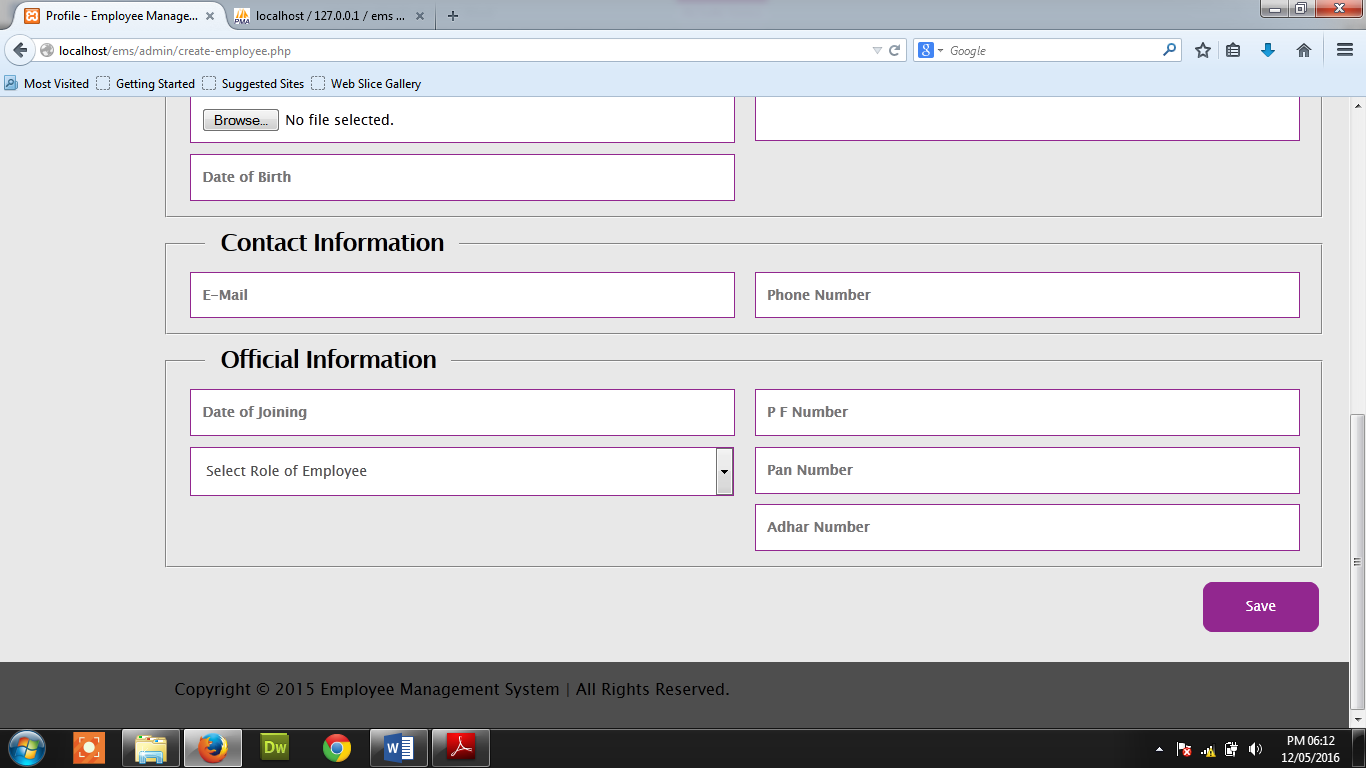
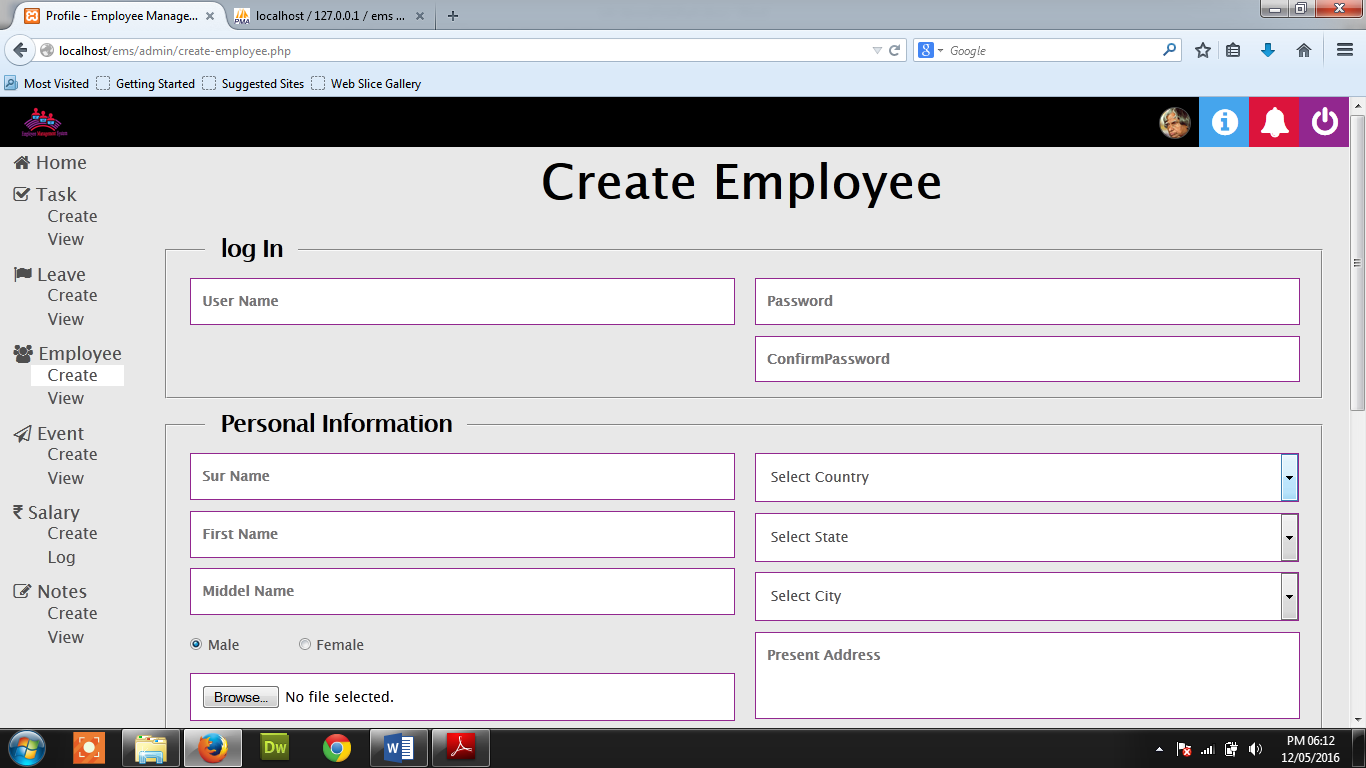
Different levels of testing are used in the test process; each level of testing aims to test different aspects of the system.

The first level is unit testing. In this testing, individual components are tested to ensure that they operate correctly. It focuses on verification efforts.

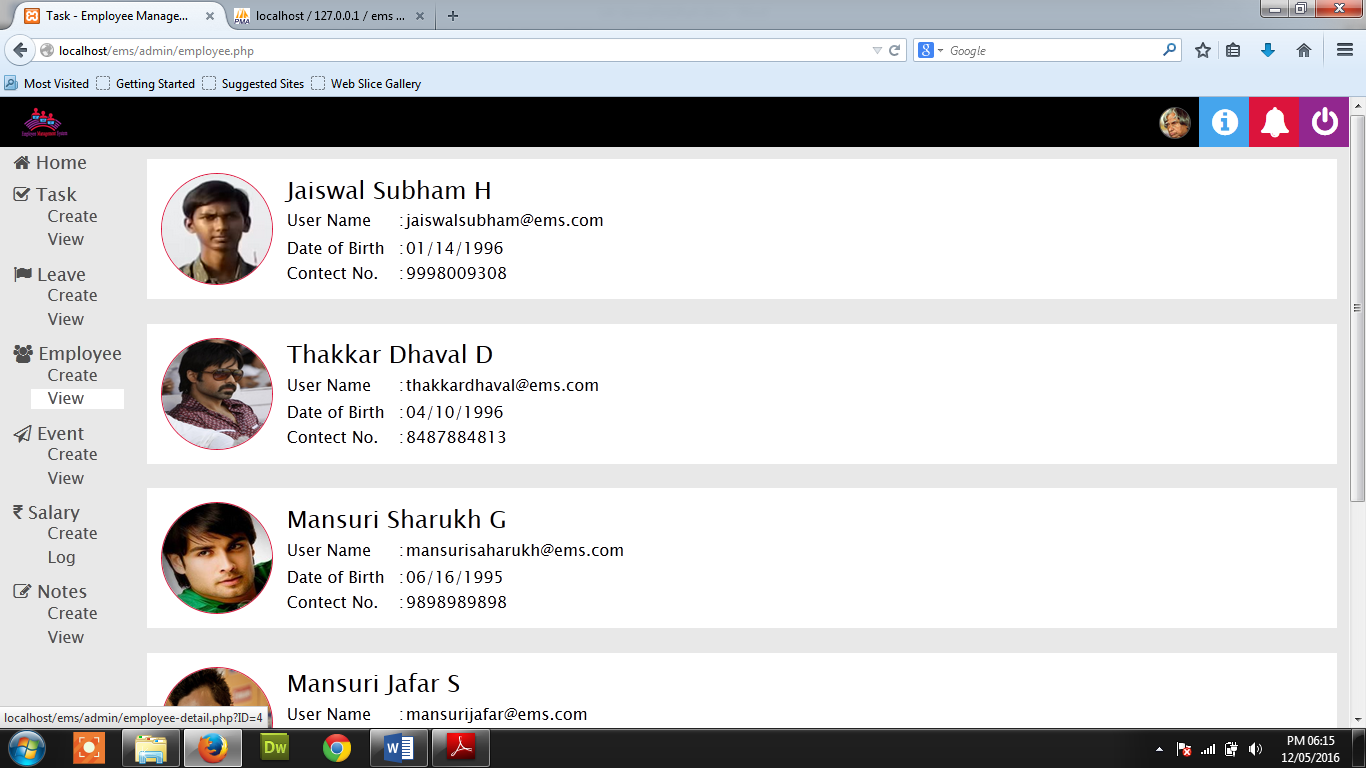
The second level is integration testing. It is a systematic technique for constructing the program structure. In this testing, many tested modules are combined into the subsystem which is then tested. The good here is to see if the modules can be integrated properly.

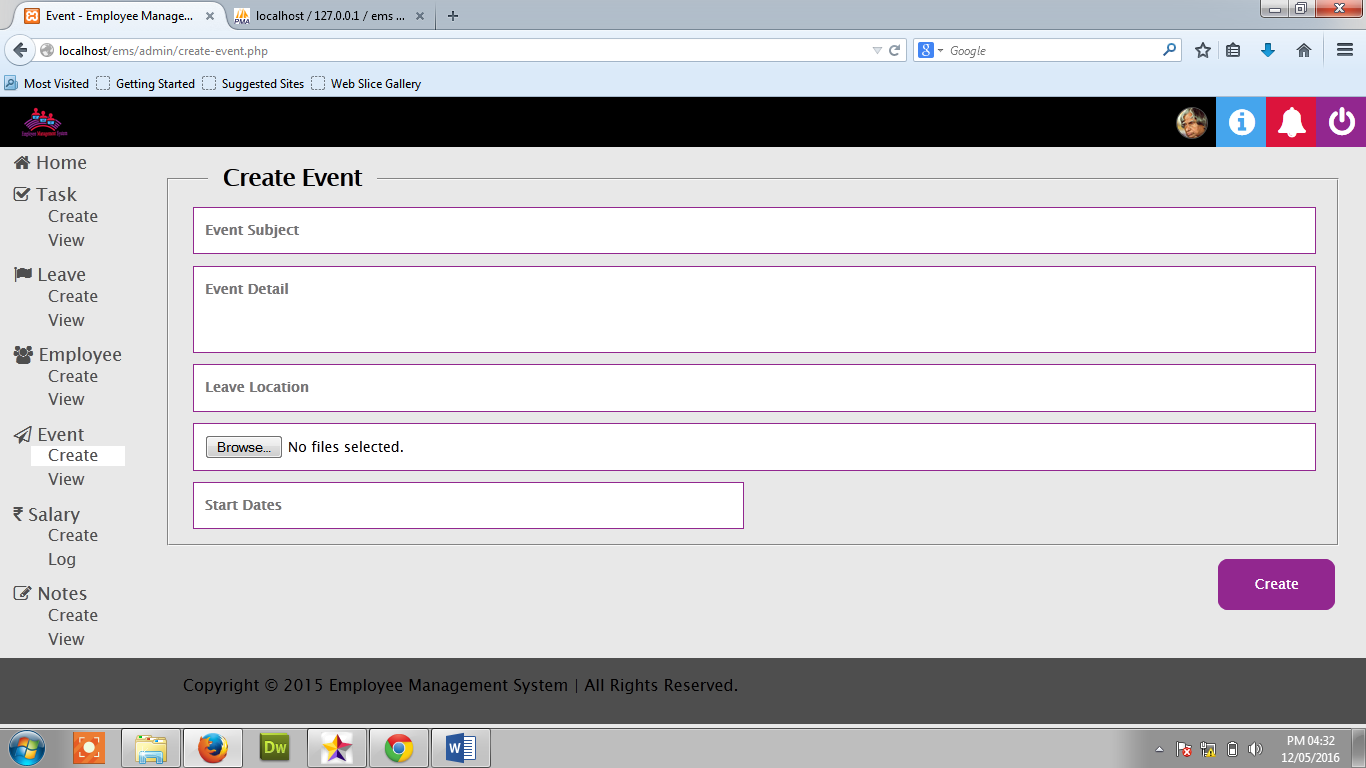
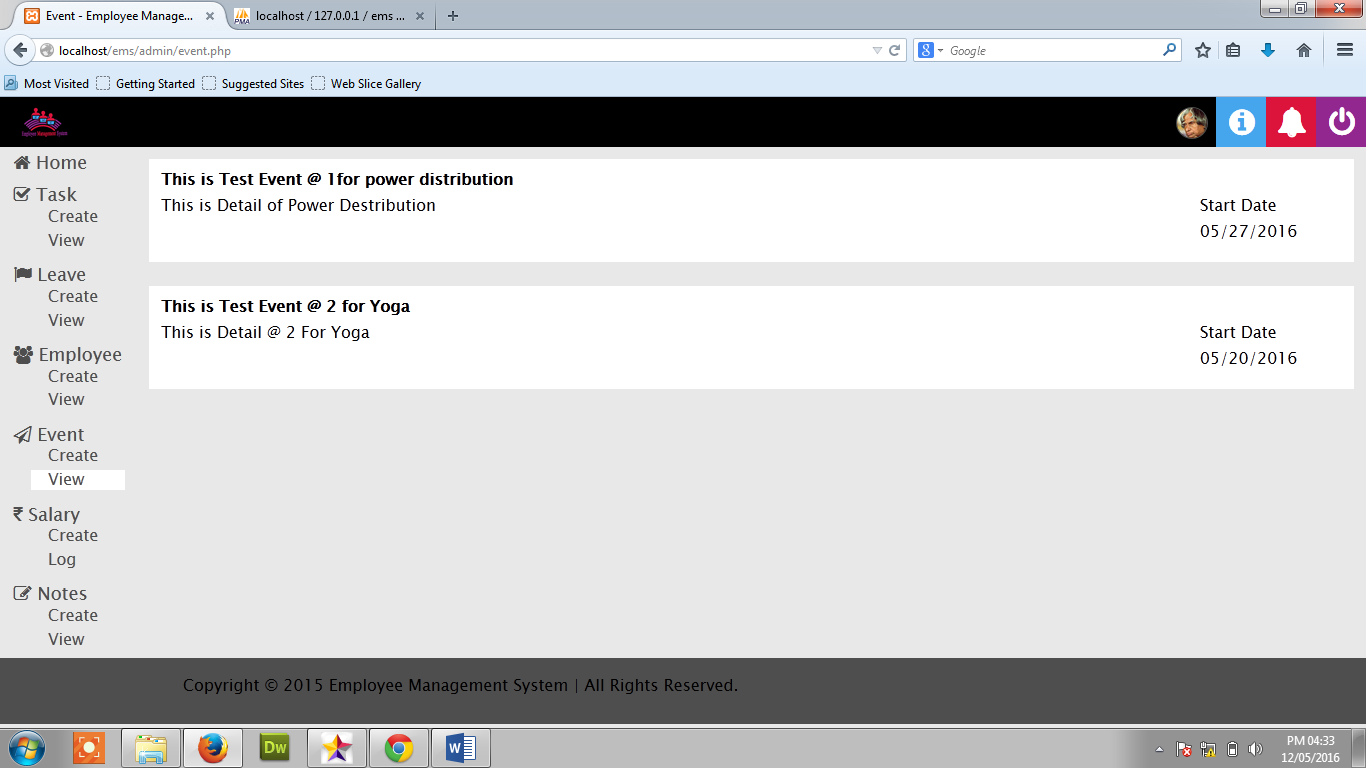
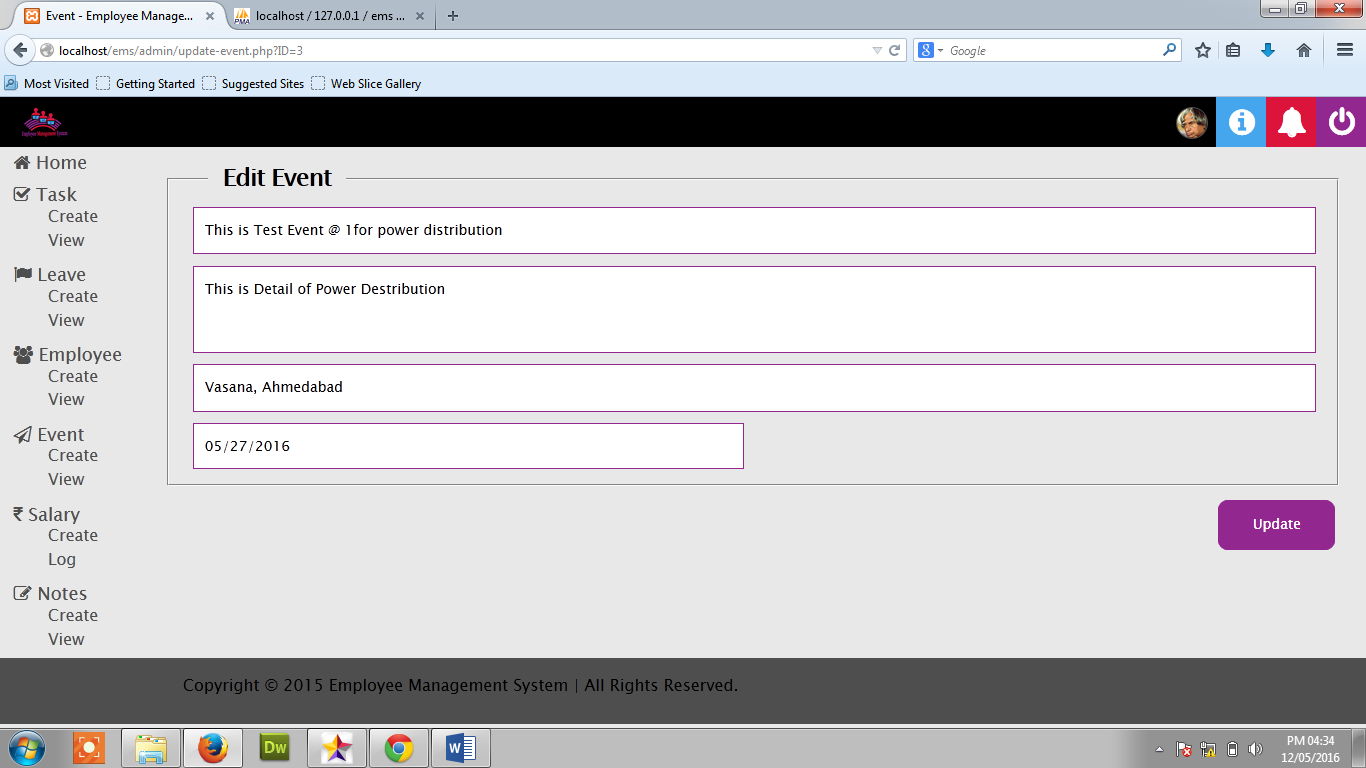
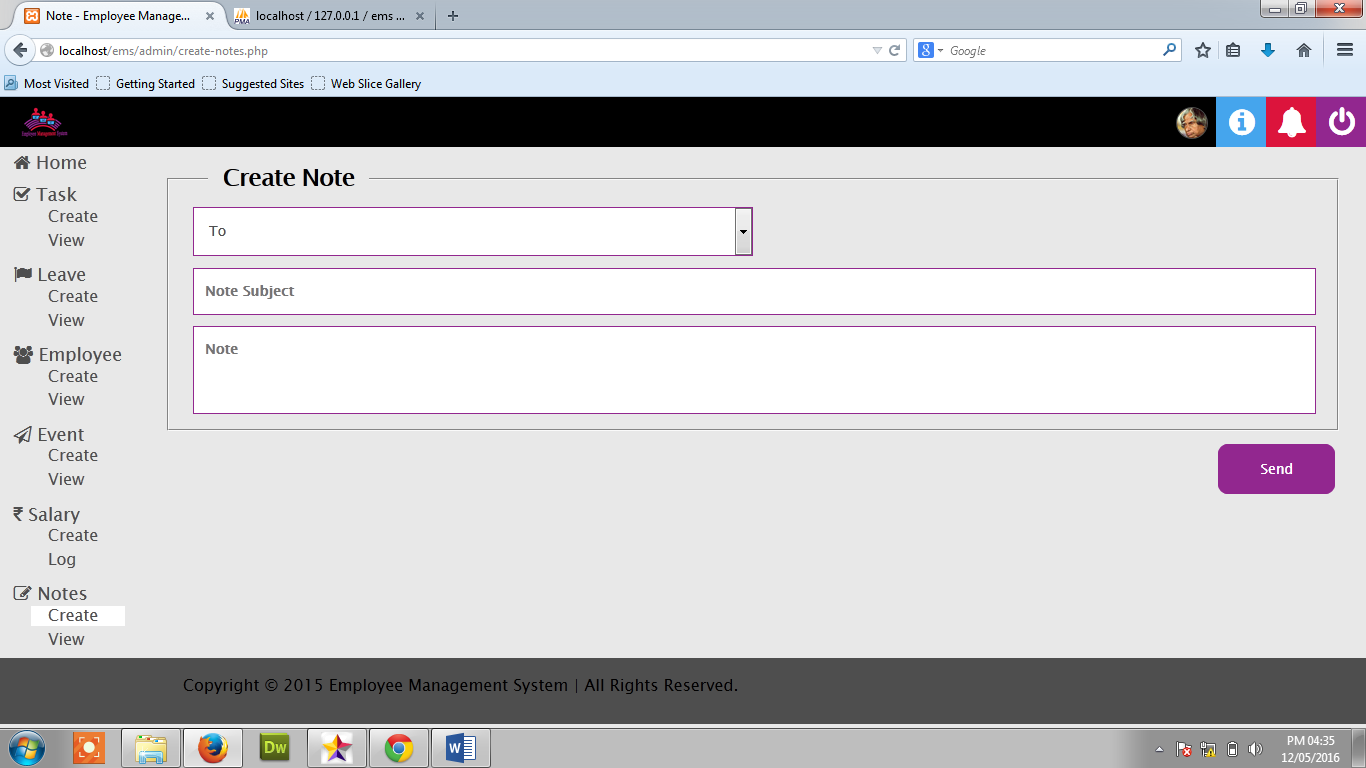
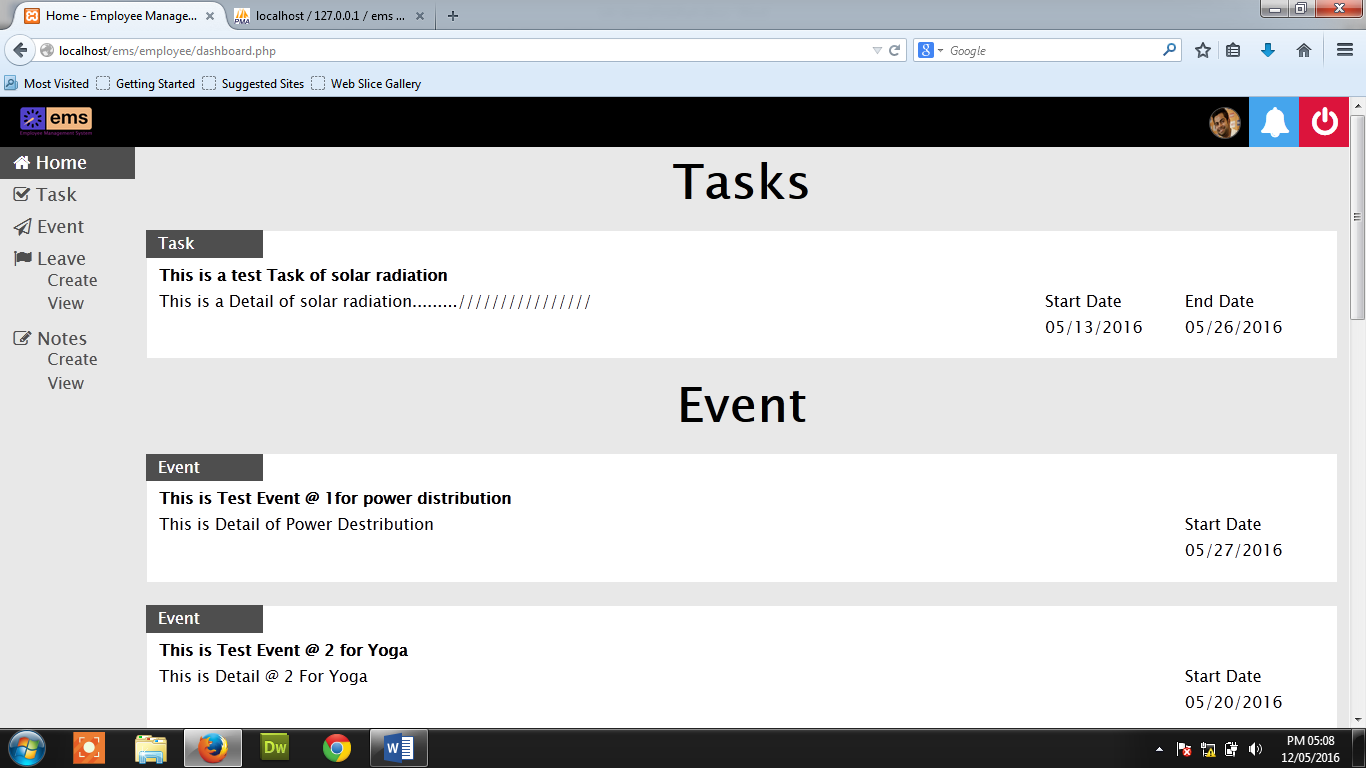
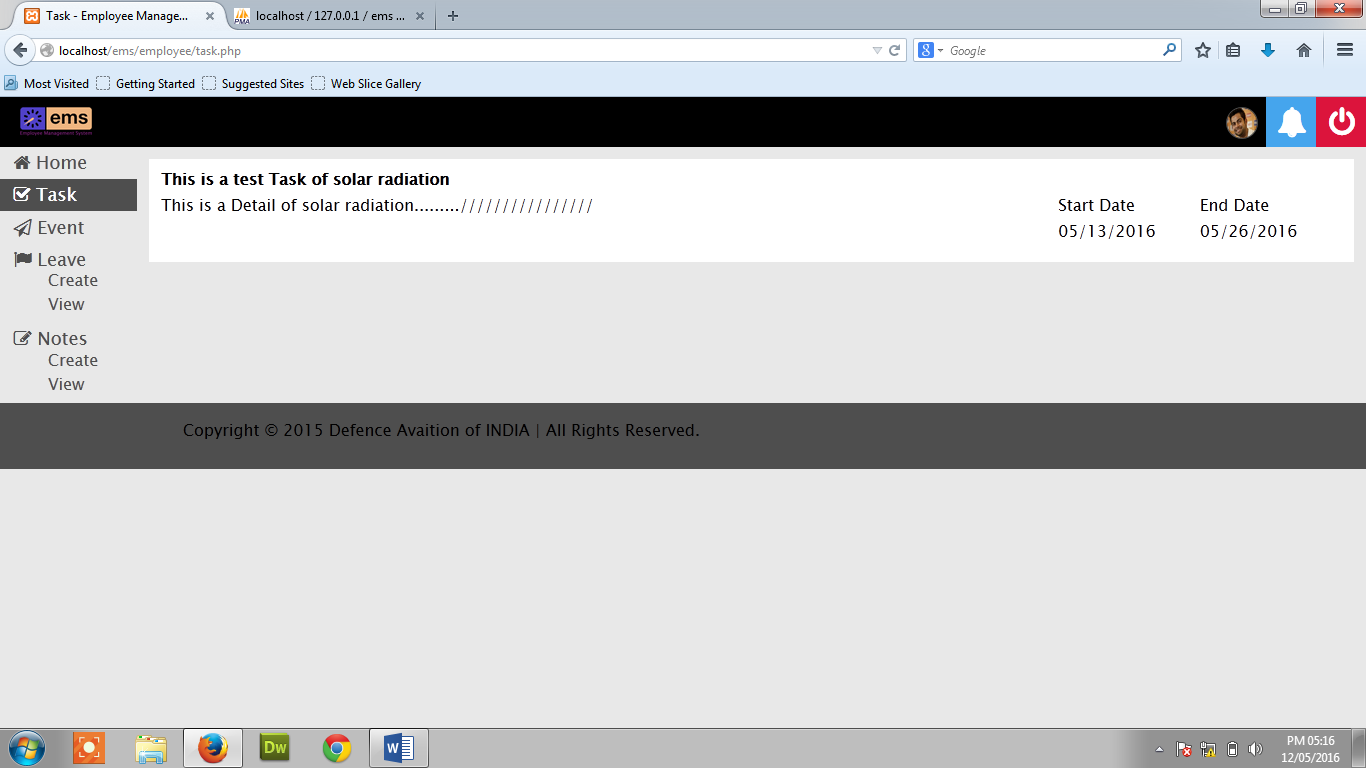
The Third level is integration testing. System testing is actually a series of different tests whose primary purpose is to fully exercise computer based system. These tests fall outside scope of software process and are not conducted solely by software engineers

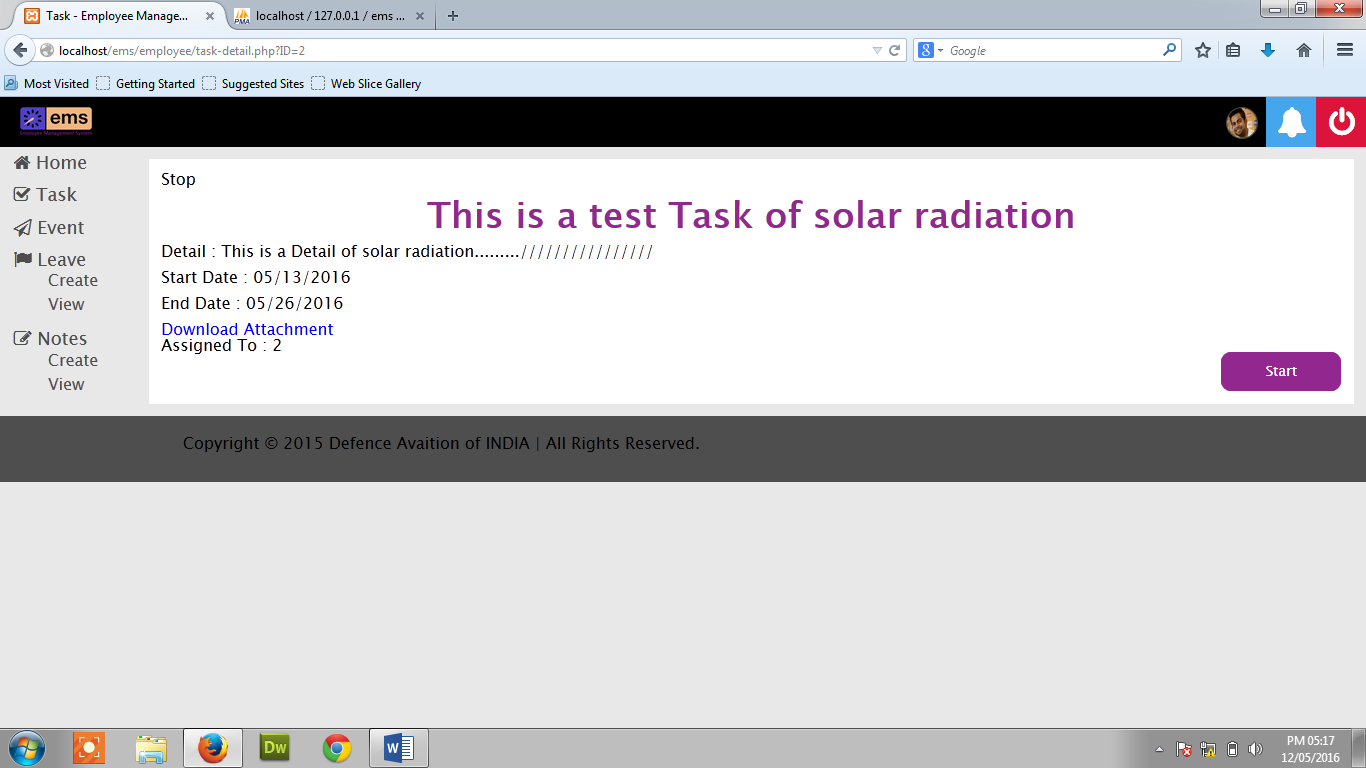
1. **Screen Short**
   1. **Login Page**
   2. **Deshboard Page of Admin side**
   3. **Create Task Admin Side**
   4. **View Task Admin Side**
   5. **View Task with full Detail Admin Side**
   6. **Edit Task Admin Side**
   7. **See Leave Request of Employee at Admin side**
   8. **Employee Leave Approve & Reject**
   9. **Create Employee at Admin Side**

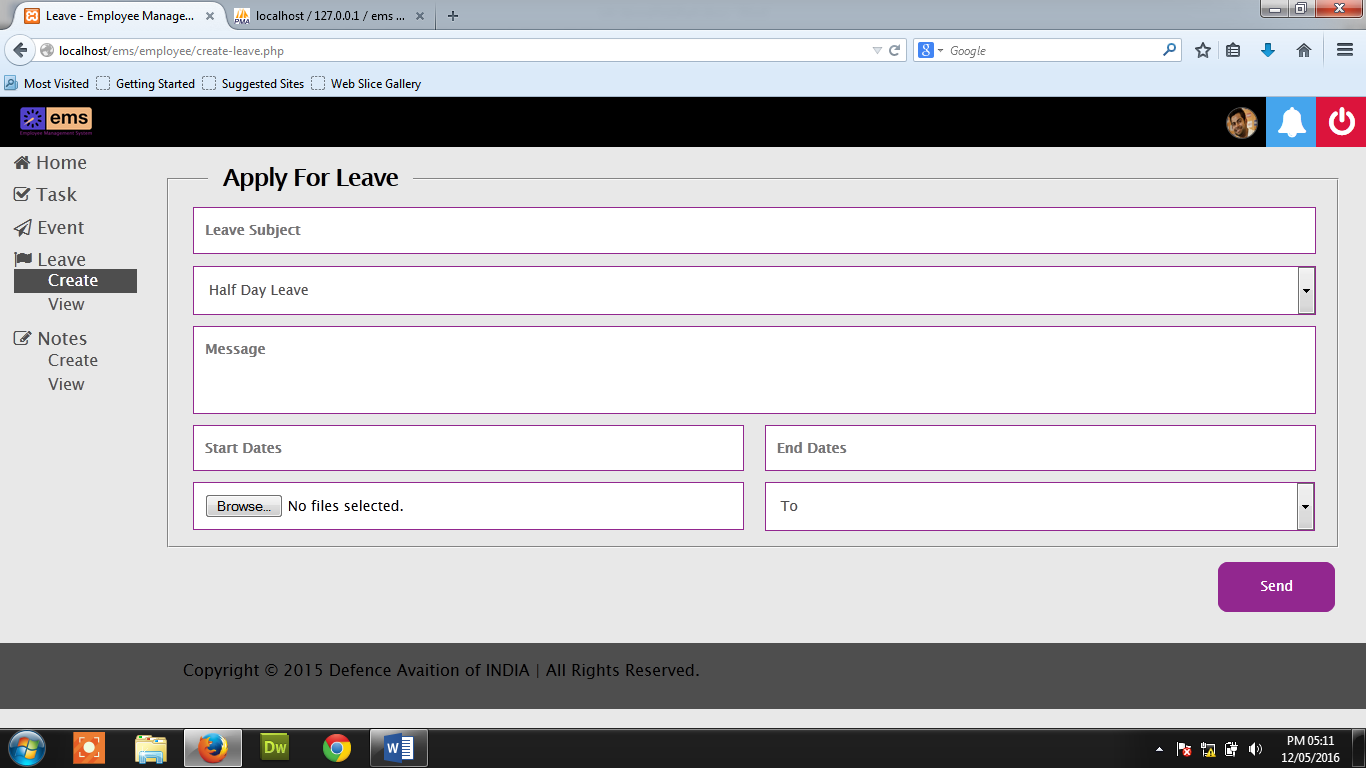
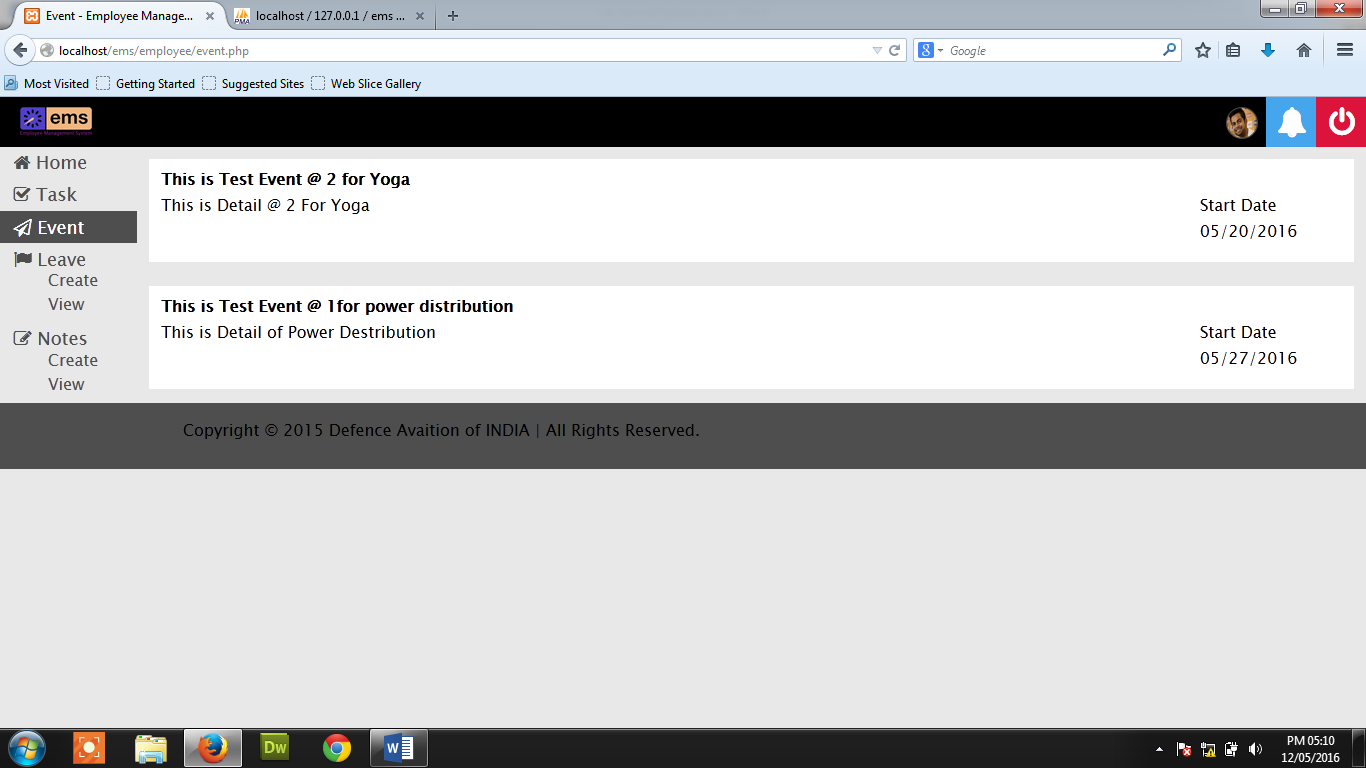
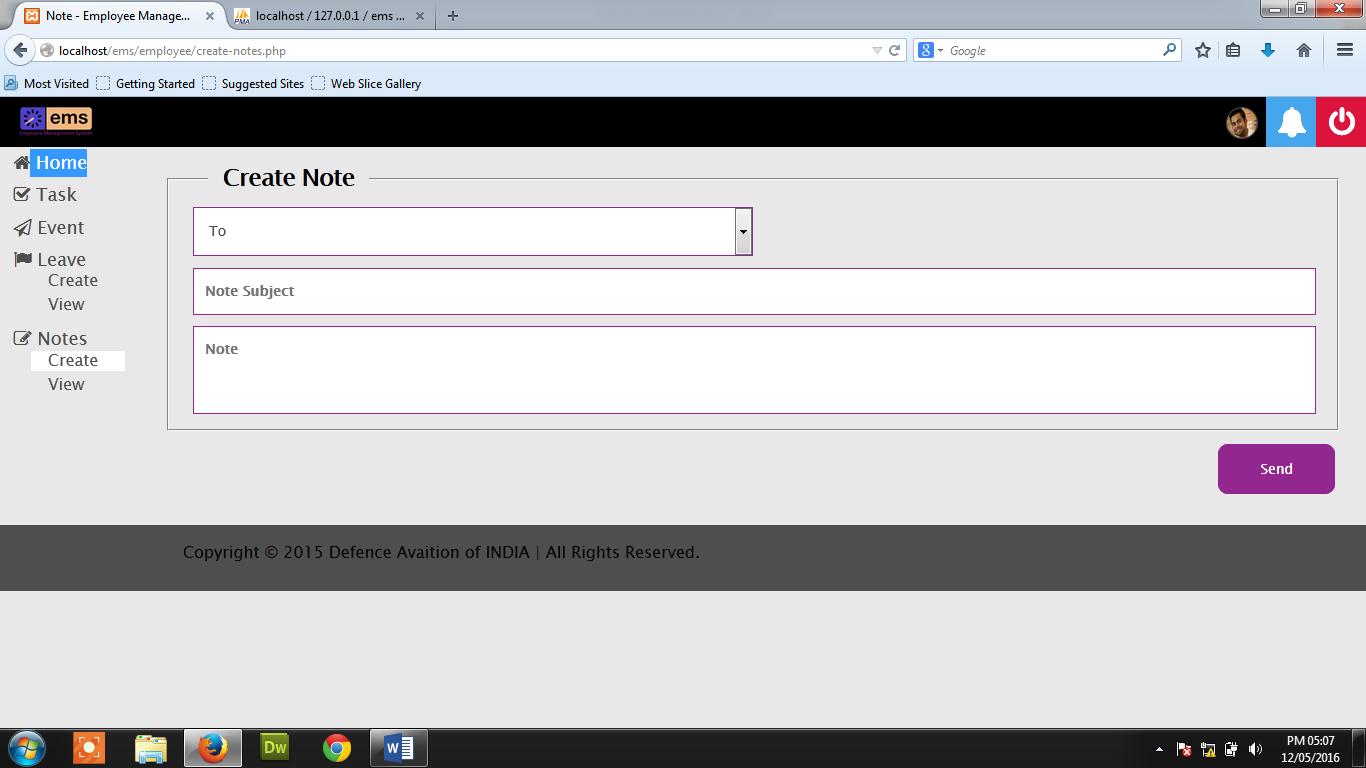


* 1. **View employee Admin Side**



* 1.  **Create Event Admin Side**
  2.  **View Event Admin Side**
  3.  **Edit Event Admin Side**
  4.  **Create Note Admin side**
  5. **Deshboard of Employee Side**
  6. **Task view Page of Employee Side**
  7. **Task Start & stop at Employee Side**



* 1. **Create Leave of Employee Side**
  2. **Event View Page of Employee**
  3. **Create Note Page of Employee Side**

1. **Reference**
2. OrangeHRM Open Source, For Reference of fields use in database.

[**http://www.orangehrm.com/open-source-product-features-pim.shtml**](http://www.orangehrm.com/open-source-product-features-pim.shtml)

1. jQuery user interface, For Date Picker prompt in form

[**http://jqueryui.com/datepicker/#date-range**](http://jqueryui.com/datepicker/%23date-range)

1. w3schools, For lean PHP & all

[**http://www.w3schools.com/php/default.asp**](http://www.w3schools.com/php/default.asp)

1. World Sql, for Dropdown menu database of Country, State, City

[**https://myip.ms/info/cities\_sql\_database/World\_Cities\_SQL\_Mysql\_Database\_2016\_year.htmlSS**](https://myip.ms/info/cities_sql_database/World_Cities_SQL_Mysql_Database_2016_year.htmlSS)

1. Employee Self Service, Retrieved: January 15, 2014, from: [**http://en.wikipedia.org/wiki/Employee\_self-service\_%28web-based\_application%29**](http://en.wikipedia.org/wiki/Employee_self-service_%28web-based_application%29)