# **On-Site Inspection Web App Development Report**

Table of Contents

[**On-Site Inspection Web App Development Report** 1](#_Toc109857999)

[**Software Development** 1](#_Toc109858000)

[**Software Design** 2](#_Toc109858001)

[**Wireframe** 3](#_Toc109858002)

[**Database Design** 5](#_Toc109858003)

[**Software Development Documentation** 6](#_Toc109858004)

[**User Story 1** 6](#_Toc109858005)

[**User Story 2** 8](#_Toc109858006)

[**Testing** 9](#_Toc109858007)

[**Conclusions** 9](#_Toc109858008)

[References 10](#_Toc109858009)

## **Software Development**

The site inspection is a web application. The design is developed with HTML, CSS Languages also used Bootstrap. And the backend is developed with PHP Language and Laravel framework. To prevent SQL injection it is replacing (‘) to (\‘). So, it is not getting any SQL injection attack at all.

In the include folder there has connection.php, head.php, header.php. Here connection page contains mysql connection and head contains bootstrap and normal HTML head code. And the last header page contains normal home page (while logout/new browsing) header. Here has Login, Register and the website name.

In “includes” folder there has one folder called “nav” and others php files contains navbar. These navbars are the Nav and Header of logged users/inspectors/manager/admin.

In UploadedImages folder all the profile images are uploading from website.

admin.php this file is dashboard of Admin. admin\_register.php this file is to show user management in the Website as Admin role. Here admin can change the users/managers/inspectors information. Also here admin can create another user/inspectors/managers/admins. From this register\_edit.php file admin can edit the user/inspectors/managers/admins profile information.

index.php this file is the Login page while logged out. logout.php this file is to logout the current role user. manager.php this file is the dashboard of Manager role. site\_inspector.php this file is the dashboard of Site Inspector. site\_inspection\_form.php this file is for the Site Inspection Form page for the site inspector role. Site Inspector can see all the inspections in this file site\_inspector\_All\_Inspections.php. And he can edit the inspection form from site\_inspection\_edit.php this file.

user.php this file is the User Dashboard. On user\_account.php this file user can change his profile information. Also he can update his profile picture on this page.

### **UI**

The application has 5 steps which as depicted in the pictures below include:

1. Add company details

2. Add sites

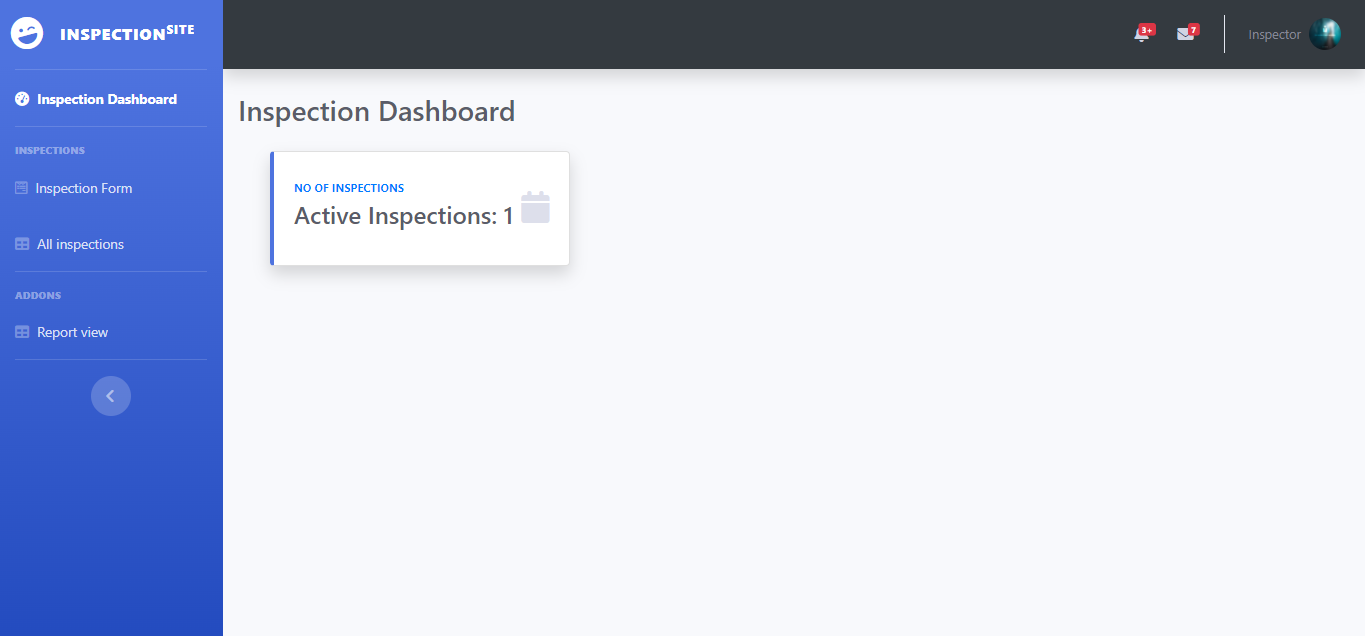
3. Add site inspectors

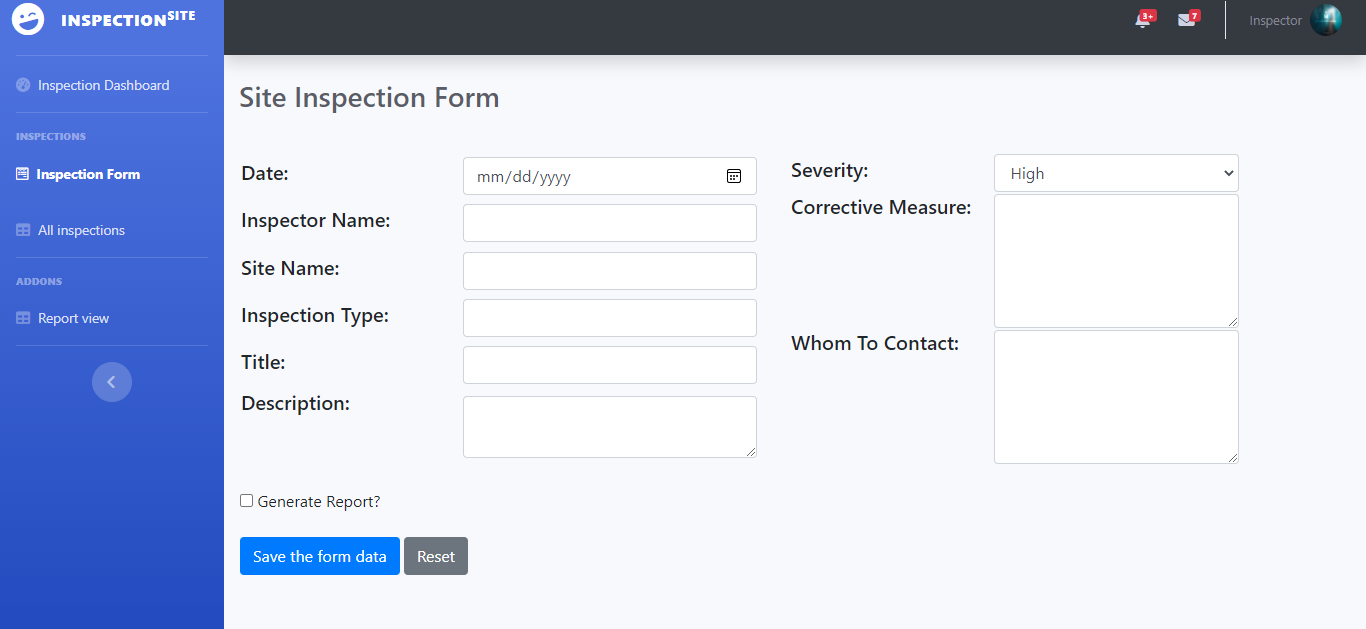
4. Create an inspection form

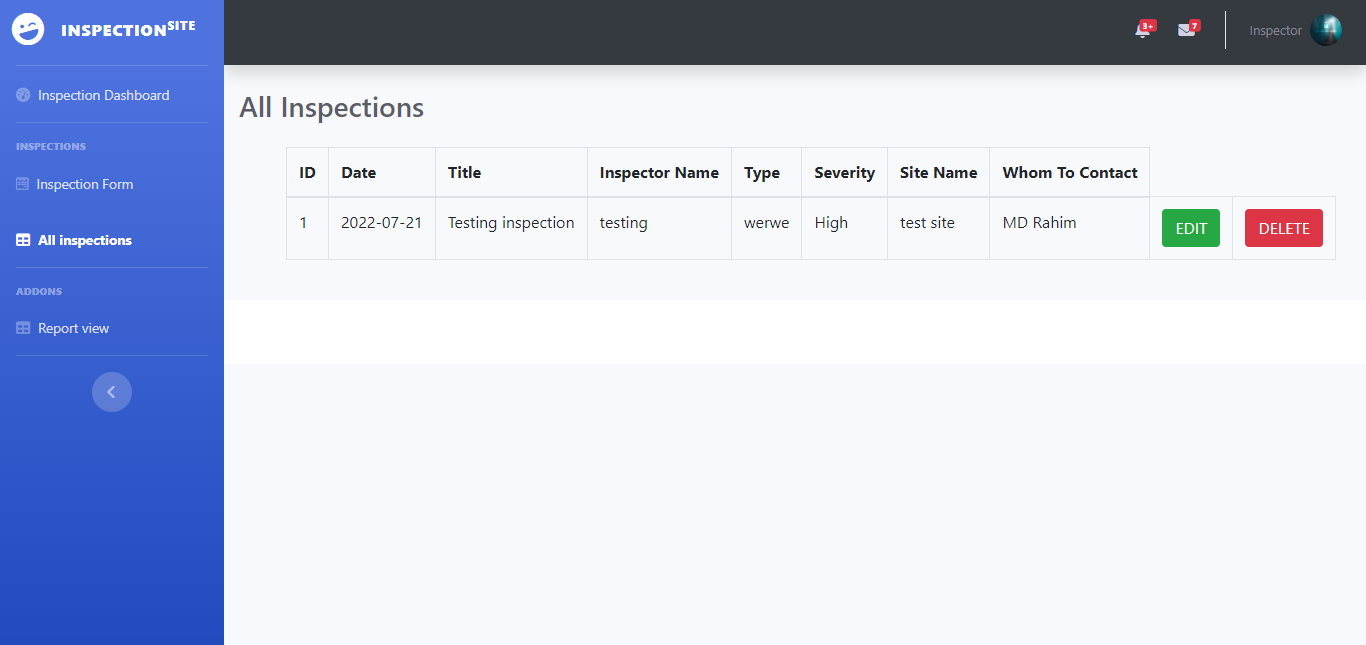
5. Send mail notification about inspection report

There are 4 user roles which are: Site Inspector, Admin, Manager, and User. Only Admin can register others person. Here site inspector can create multiple Site Inspection Forms. There has inspection date, Inspector Name, Site Name, Inspection Type, Title, Description, Severity (High, Medium, Low), Corrective Measure, Whom To Contact fields. All the forms can be seen at once in all inspections tab.

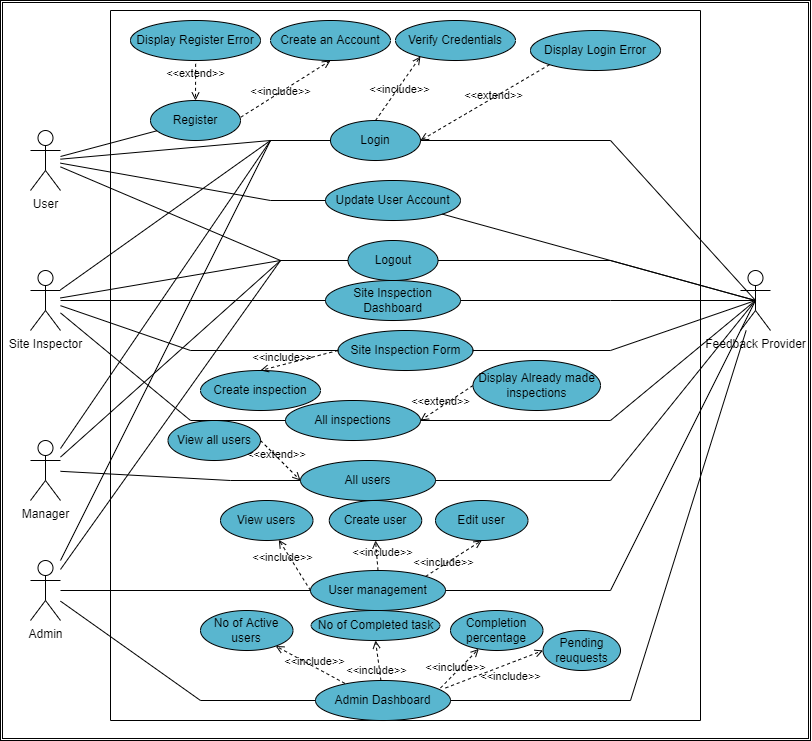
He can also edit and delete inspections form there. In the site inspector dashboard, there has total number of inspections.







## **Software Design**



User interaction (UI) interfaces or views are constructed making use of pre-installed templates on ZoHo creator as it serves as the basis for a plethora of app development services. The invocation of built-in APIs and the communication between views and app services are both implemented by Bootstrap.

In other words, Bootstrap is used to send events that are triggered by user input to built-in APIs and App services, and Bootstrap is also used by App services and built-in APIs to supply data to views. In this approach, ZoHo creator makes a number of application programming interfaces (APIs) available for access to a variety of devices, including cameras, GPS sensors, audio, local storage, and the network.

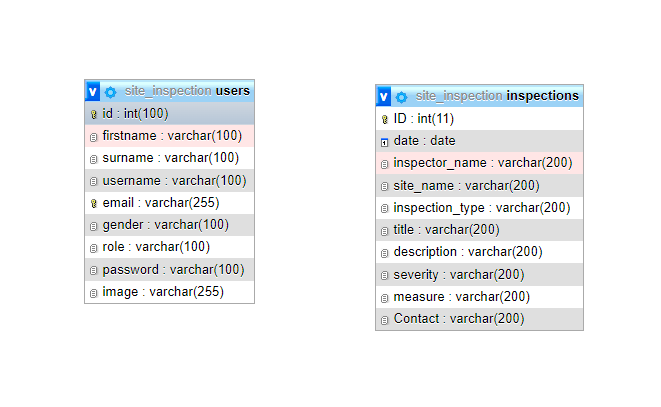
Additionally, because of the simplicity of the platform, HTML5, and CSS was easy to use as the platform these languages for website app development services, it is quite straightforward for web developers to create micro-Apps.

The ZoHo creator platform possesses a number of capabilities, including as GPS sensing, audio and video recording, reading QR codes, and uniform application programming interfaces (APIs) for developers, that reduce the amount of effort required for the development process. The performance of HTML5-based online applications is typically worse than that of native Apps, whereas the performance of micro-Apps is typically on par with that of native Apps.

This is because ZoHo, which has worked hard to improve the performance, has integrated all UI elements and their styles. When user interfaces and designs similar to those seen in ZoHo are utilized, it is easier for users to comprehend and use the application.

To prevent SQL injection, it is replacing (‘) to (\’). So, it is not getting any SQL injection attack at all. Also, for log of Testing done, PHP echo gave a lot help during building the website. Whenever I was not getting expected values then I was using PHP echo. Such as while registering users POST, inspection form saving, editing inspections, editing registered users, also sometimes used it to check SQL query for SQL injection and so on. When website is finished, I removed those log codes.

### **Database Design**



MySQL database is used for database site inspection. Two tables are present. They are inspections and users. Id, first and last names, usernames, emails, gender, roles, and passwords are all listed in the users table. Additionally, the inspections table includes the following information: ID, date, inspector's name, site name, inspection type, title, description, severity, measure, and Contact. On localhost, the MySQL database is now active. Additionally, include/connection.php in the code contains the database connection.

Each role's ID, Name, and perhaps additional characteristics should be listed in the table of roles so that we can grant each role access. The table role rel user, where each row's UserID and RoleID columns are filled with the matching IDs for each user and role, is the last place where the relationship between users and their related roles is recorded.

In order to distinguish between records for safety inspection and quality inspection, TypeID is a flag utilized. In the interim, the priority level, such as common or most important, and the status, such as pending or resolved, are stored in the columns LevelID and StatusID. The CreatedBy, OperatorIDs, and CC columns contain information on the creators, possible users who might be responsible for the problem, and users who should be informed of it.

These tables make it possible to properly gather and store data for users and on-site quality and safety assessment. The compilation of issue reports and further data analysis will help the construction management decision-making process.

## **Testing**

The web application for on-site inspection was put through its paces by being tested with a number of different user logins, as well as by running a variety of on-site simulations, all of which resulted in good and immediate responses.

## **Conclusions**

The intuitive website application was developed specifically for the purposes of site inspection and documentation. It is designed to be successfully utilized for site inspection throughout both the construction phase and the operation phase of the project. A case study that involved the inspection of a building’s envelope was carried out with the assistance of ZoHo.

Because of this inspection, it was determined that on-site inspection website application makes it simpler for engineers and managers to record information regarding problems, the people involved, and the locations of defects using GPS. Inspectors are able to generate reports and papers with specific information and formats for a range of uses. These reports and papers can be used for a variety of purposes.

Site engineers and managers may be able to save a significant amount of time and prevent errors by utilizing the automated documentation and reporting process provided by InSite Inspector. These benefits are in contrast to the outcomes of using standard site inspection processes. In further studies, the precise position of the defects will be improved, and techniques using image processing and machine learning will be utilized to automatically classify any defects that are found on-site.

Because facility owners can exercise tight control over their staff and heavily customize the mobile platforms used by their escorts, these technologies can be used in a much wider range of situations for on-site inspections. Even in very sensitive areas, it is not impossible for facility employees to carry mobile computing devices into the building in a safe and secure manner if enough administrative and engineering precautions are implemented.

On the other hand, inspection parties are obligated to comply with the often-stringent regulations that are outlined in treaties or agreements. Under the current on-site inspection regulations, there are very few options available for mobile computing devices that can be utilized in inspection team activities. Inspectors believe that the most beneficial application of these technologies may be in the realm of training, as well as in making the most of the time spent traveling to and from work places.

On inspection visits for agreements that are currently in effect and have stringent equipment standards, it is highly doubtful that mobile devices will be allowed to be brought along. This does not prevent the possibility that mobile devices such as cellphones, tablets, and wearable systems could be helpful additions to the information and communication systems utilized by the parties in question at an offsite operations center or any other place.

Mobile technologies have the potential to be of great assistance in circumstances such as these since inspection preparation and training tasks are performed away from the actual location of the inspection.

# References

DTIRP. (2008). "The Arms Control Inspector." Product No. 406P. . *DTIRP Outreach*.

Gough, R., & Alan, Z. (1993). *Arms Control Compliance: Information Value of Verification Measures. SAND93-1347C.*

Lundkvist, R., Meiling, J. H., & Sandberg, M. (2014). A Proactive Plan-Do-Check-Act Approach to Defect Management Based on a Swedish Construction Project. *Construction Management and Economics, 32(11)*, 1051-1065.

Park, C., Lee, D., Kwon, O., & Wang, X. (2013). A Framework for Proactive Construction Defect Management Using BIM, Augmented Reality and Ontology-based Data Collection Template. *Automation in Construction, 33*, 61-71.