## INTRODUCTION

The goal of this project is to use PHP technology to build and implement an aircraft management system with a user interface, administrator interface, and staff interface. The system also allows us to add, amend, and delete records or information about the company's services, as well as information about the company's Engineers/Mechanics. All of this data is saved in a MySQL database, which employs encrypted passwords to ensure the data's security. We use the PHP Language's file-handling feature to store and retrieve data. It allows us to add as many records as we want to our database. Because records are searched sequentially, retrieving information from a database is significantly faster when we search for a specific record.

### About Project

The "Aircraft Management System" project is written in PHP, which is well-suited to this type of application. It is the most powerful tool for system development due to its user-friendly design, in-built documentation, intricacy, error detection, binding facilities, and interface with other system packages. PHP also includes all of the technologies that aid in the development and operation of dependable, scalable systems.

The user is assisted at every step of the way, ensuring that there are no issues encountered when utilizing it. Furthermore, the report's descriptions of each step, as well as the user manuals, make it very straightforward to comprehend. The software and report have been meticulously designed to be clear, easy, and error-free, which is what distinguishes them.

## Objective of the System

The objective of the Aircraft Service Management System  includes:

## Provide better and more secure information to the system's users in order to improve the system's performance in providing services.

## To give some automation in aviation maintenance management.

## .To assist aviation service management in increasing the efficiency of their operations.

## It will also demonstrate the management's attitude that they are aware of the newly discovered information.

## Technology has been introduced, and people are eager to adopt it.SYSTEM SPECIFICATION

### **Hardware Requirements**:-

* Pentium-IV(Processor).
* 256 MB Ram
* 512 KB Cache Memory
* Hard disk 10 GB
* Microsoft Compatible 101 or more Key Boar

### Software Requirements:

* Operating System :           Windows
* Programming  language:   PHP
* Front-End:                          HTML, CSS, JAVASCRIPT
* Back-End:                           SQL
* Web Server:                       Apache Server

## FEASIBILITY STUDY

**A feasibility study is carried out to select the best system that meets performance requirements.**

To choose the optimal system that meets performance requirements, a feasibility study is conducted.

The term "feasibility" refers to the process of determining whether or not a project is worthwhile. A feasibility study is the procedure used to arrive at this conclusion. This form of research determines whether or not a project can and should be pursued. Because the feasibility study may result in the commitment of significant resources, it is critical that it be completed effectively and without fundamental mistakes of judgment. The survey is expanded to a more extensive feasibility assessment based on the findings of the original inquiry. A feasibility study evaluates a system concept for workability, organizational impact, ability to meet user needs, and effective use  of resources. The goal of the feasibility study is to get a sense of the problem's scale rather than to solve it. The problem definition is solidified during the study, and aspects of the problem that will be included in the system are determined.

## SYSTEM ANALYSIS

One of the most significant phases in the system development life cycle is the system analysis phase. It is critical that the software developer conduct a thorough analysis of the existing system. A thorough analysis of the system is conducted, with users' needs (i.e. features that are crucial to the system's success) and wants (i.e. features that would be nice but aren't essential) being identified. The developer will be able to understand the complexities of the present system thanks to the research.

## **SYSTEM DESIGN**

Design is the initial step in the development phase for any methodologies and ideas used to define a device, a process, or a system in enough detail to allow physical implementation.

Following the analysis and specification of software requirements, software design entails three technical activities: design, coding, implementation, and testing, all of which are necessary to develop and verify the software.

The design activities are critical in this phase since they are where decisions are made that will ultimately effect the success of the software implementation and its ease of maintenance. These choices have the most impact on the system's dependability and maintainability. Only via design can the customer's requirements be accurately translated into finished software or a system.  
In development, design is where quality is fostered. Software design is the process of converting requirements into a visual representation of software. There are two stages to software development. The transformation of requirements into data is the focus of preliminary design.

### PETRI MODEL DIAGRAM

Petri nets are well-known models for describing systems with concurrency, synchronization, and randomness. Wang et al [19], Murata [10] define a Petri net as a directed bipartite graph with two types of nodes: locations and transitions. Directed arcs link the nodes together. It is not possible to connect two nodes of the same type. Places are represented by circles in graphical representation, and transitions are represented by bars or boxes.

The state transition diagram of the system is given in the figure below.

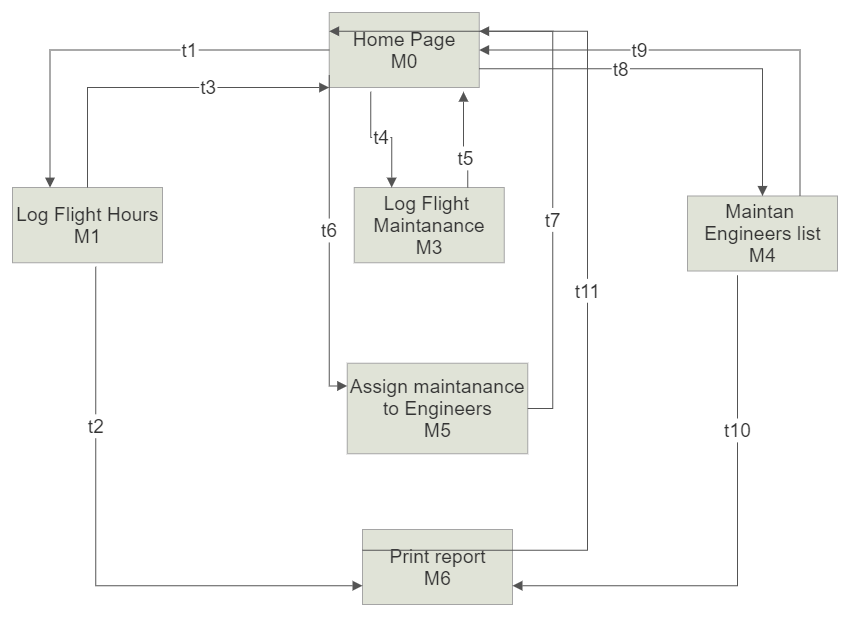
****

Figure State transtition diagram

M0 M1 M5 M0 (t1 ●t2●t3)

M0 M2 M0 (t4● t5)

M0 M3 M5 M0 (t6●t7●t3)

M0 M3 M0 (t6 ● t8)

M0 M4 M0 (t9●t10)

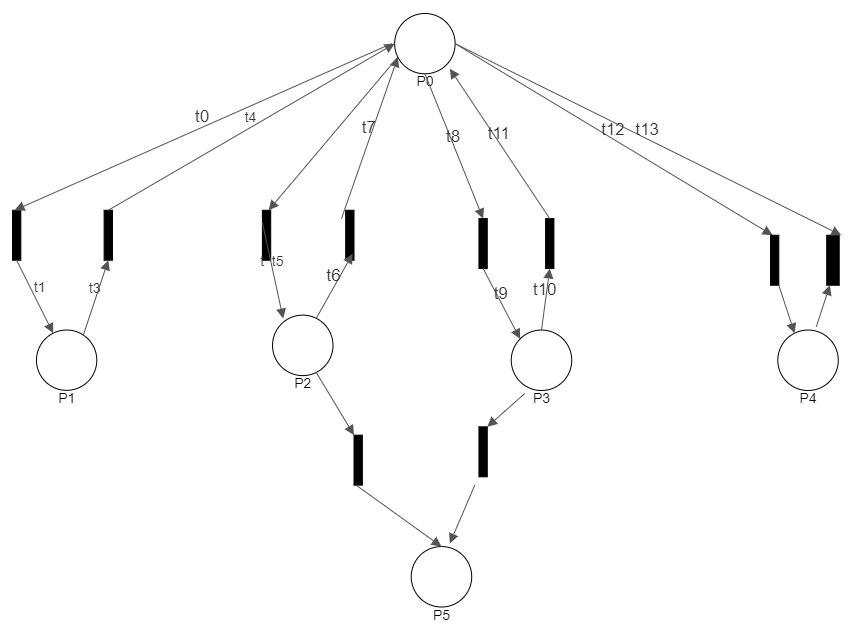
By simply making each state a place and each arc between two places a transition, the graph shown above may be turned into a comparable Petri net. The transformation's outcome is illustrated below.****

Figure Petri Model diagram

### USE CASE DIAGRAMS

### 

Administrator

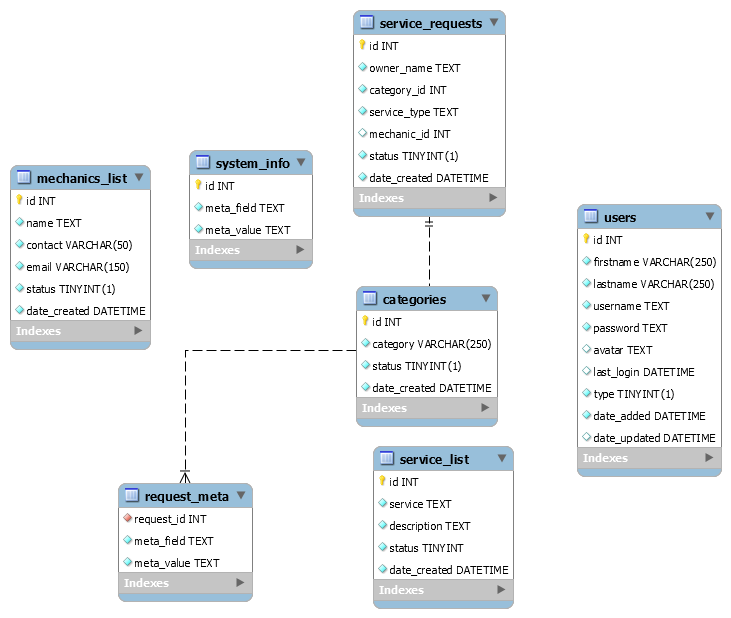
Client

Staff

### E-R DIAGRAM

In 1976 [Chen76], Peter presented the Entity-Relationship model as a way to unify network and relational database views. Simply said, the ER model is a conceptual data model that views the real world through the lens of entities and relationships. A major component of the model is the Entity-Relationship Diagram, which is used to visually represent data elements. The concept has been expanded since Chen's work, and it is now frequently used in database design. Because it is an excellent fit for the relational model, the ER model is important for database designers. The constructs of the ER model are simply converted into relational tables. It's simple and easy to use, and it only takes a few minutes to get started. As a result, the model can be used by the database designer to express information. Furthermore, the model can be utilized by a database developer as a design strategy for implementing a data model in a certain database management program.

#### **CONNECTIVITY AND CARDINALITY**

One-to-one, one-to-many, and many-to-many are the three main types of connectedness for relations. When only one instance of entity A is associated with one instance of entity B, this is referred to as a one-to-one (1:1) relationship. A one-to-many (1:N) connection exists when there are zero, one, or many instances of entity A for each instance of entity B, but only one instance of entity A for each instance of entity B. When there are zero, one, or many instances of entity A, there are zero, one, or many instances of entity B, and when there are zero, one, or many instances of entity B, it is referred to as a many-to-many (M:N) relationship. The mapping of connected data is described by the connectivity of a relationship. For the sake of our case, we use ER below

## **SYSTEM TESTING AND IMPLEMENTATION**

Testing is the act of running a program in the hopes of discovering an error. Testing is an important part of software quality assurance since it provides the final assessment of the specification, design, and coding.

The phase of system testing is crucial. Testing is an unusual occurrence in the software world. As a result, before the proposed system is ready for user acceptance testing, it goes through a set of tests.

A excellent test case is one that has a high chance of uncovering an error that has yet to be detected. A successful test is one that uncovers a previously unknown flaw.

Testing Objectives:  
 Testing is the process of running a program with the goal of detecting errors.

* A good test case is one that has a chance of uncovering an error that has yet to be detected.
* A successful test is one that reveals a previously unknown flaw

**Testing Principles**

* All tests should be able to be linked to the needs of the end user.
* Long before the tests begin, they should be designed.
* Testing should begin on a modest scale and work its way up to large-scale testing.
* It is impossible to do exhaustive testing.

To be most successful, testing should be done by an impartial third party.

The main goal of test case design is to come up with a set of tests that have the best chance of exposing software flaws. Two different types of test case design methodologies are employed to achieve this goal. They really are.

1. White box testing.  
 2. Black box testing.

**Testing strategies**

Low-level tests that verify that all little source code segments have been appropriately implemented, as well as high-level tests that validate significant system functionalities against client requirements, must be included in a software testing plan.  
**Testing fundamentals**  
Testing is the process of running a program with the goal of detecting errors. A excellent test case is one that has a high chance of uncovering an error that has yet to be detected. If testing is done correctly, it will reveal the software's flaws.  
  
**Testing Information flow**  
The pattern of information flow for testing. To test the process, two types of input were provided. A software requirements specification, a design specification, and source code are all part of the software configuration.  
**Unit testing**  
Unit testing is important for verifying the code created during the coding phase, therefore the purpose is to test the modules' internal logic. Important paths are checked using the complete design description as a reference to find faults within the module boundary. These tests were conducted during the programming stage. Vienna SQL was successfully tested across the board.  
**Integration testing**   
Integration testing focuses on unit-tested modules and helps to create the program structure that the design phase dictates.

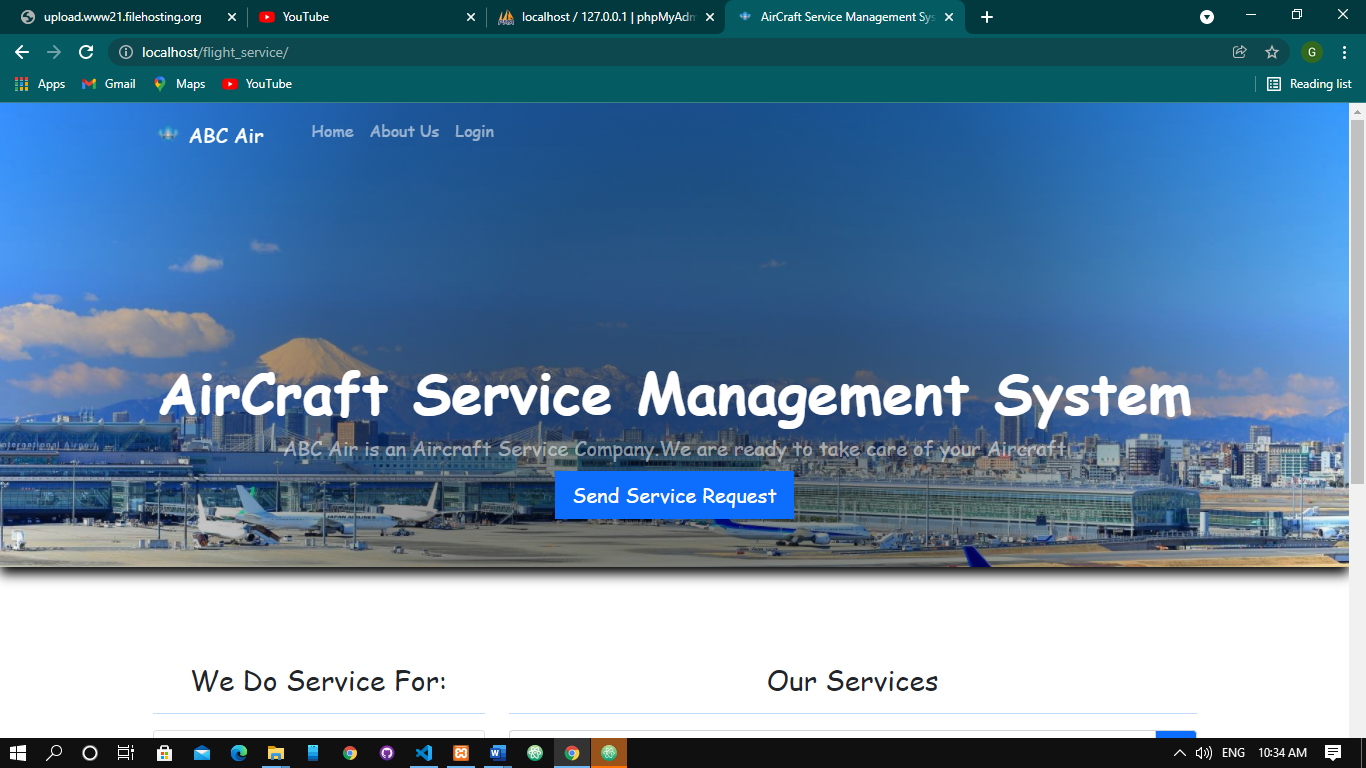
**System testing**  
System testing examines the system's overall integration. It also looks for inconsistencies between the system's current specification, initial aim, and system documentation. Individual module compatibility is the major concern. The entire system will be verified here, as well as whether the stated path ODBC connection is accurate or not, and whether the output is correct or not. These verifications and validations are performed by providing input values to the system and comparing them to expected output. Here, top-down testing is being used.

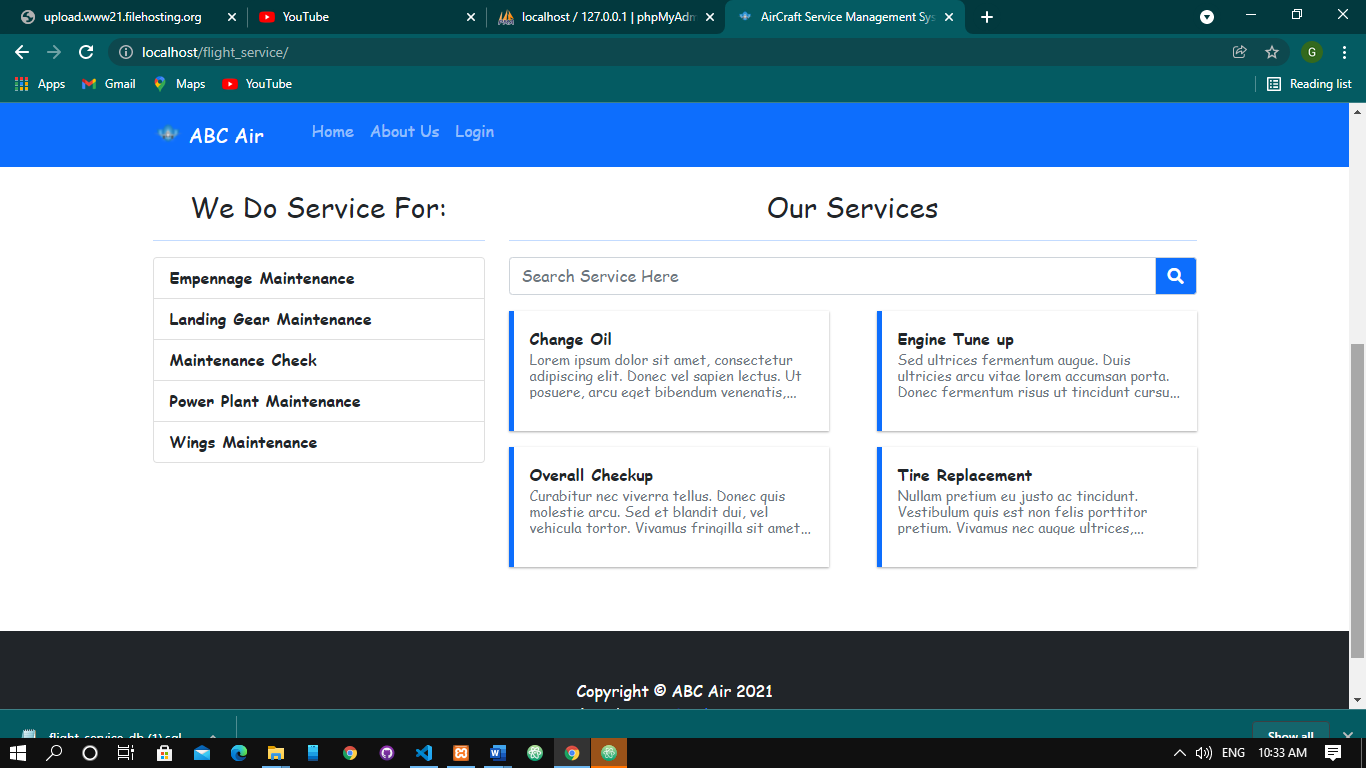
**Acceptance Testing**  
This testing is done to ensure that the system is ready for implementation. When the system is finished, acceptance testing begins. Its goal is to instill confidence in the end user that the system is ready to use. It entails the preparation and execution of functional, performance, and stress tests to demonstrate that the implemented system meets the requirements.

## OUTPUT RESULTS

### Home page

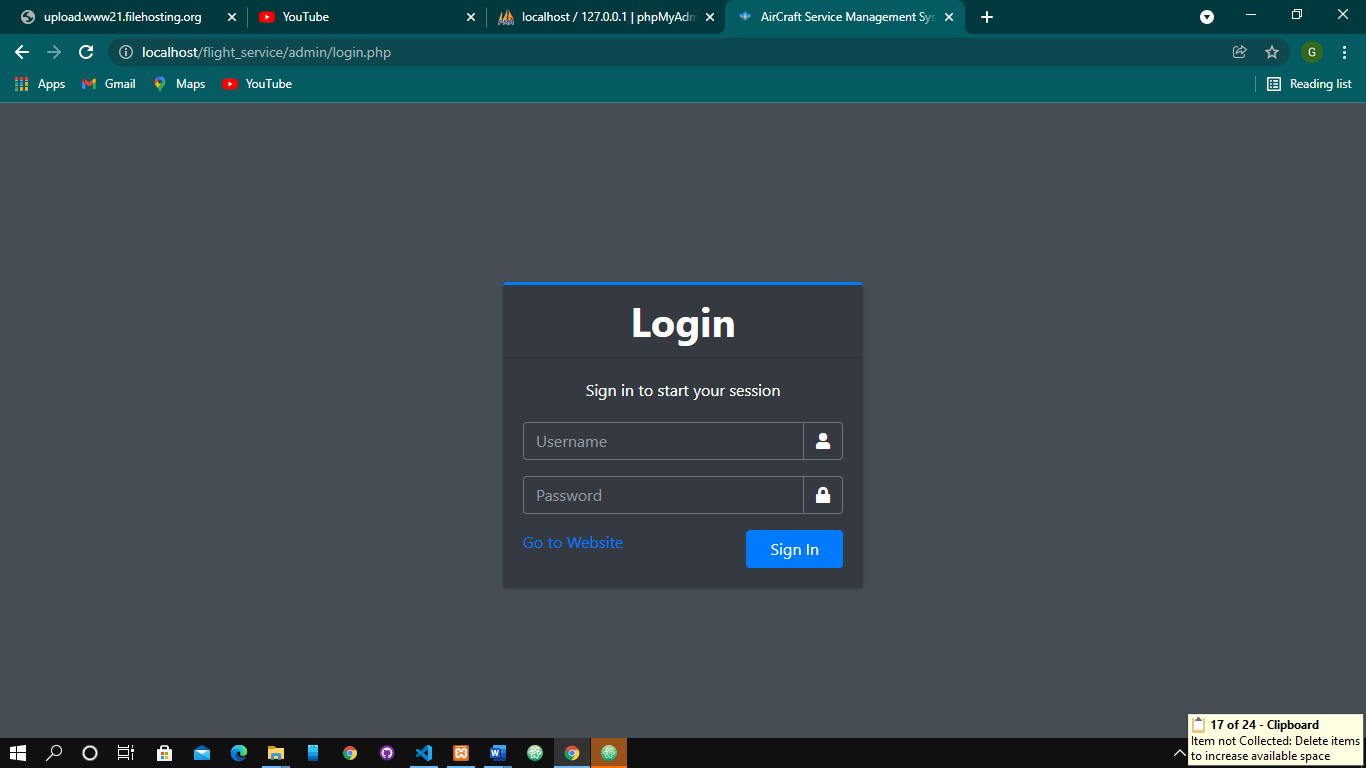
This is the user interface that first appears on running the website. Every user of the site is able to view the page whether signed in or not.





### Login page

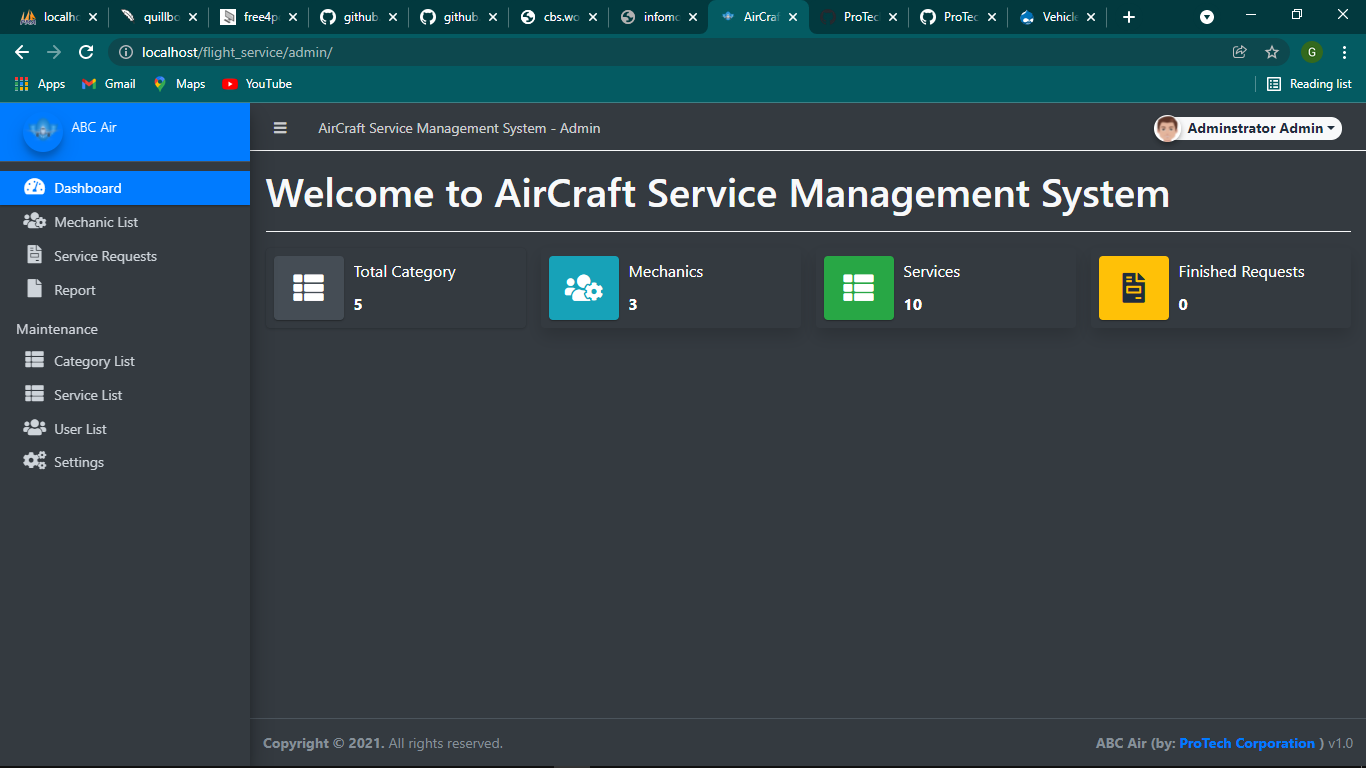
This page includes username and password field which is a requirement for logging in to the system. The password are encrypted for enhanced security.



### Administrator Dashboard

This is where everything on the website is managed from. Person in charge is the admin and manager of website. He or She is capable of

1. Adding new Engineers.
2. Adding new services to the system.
3. Assign Engineers/Mechanics tasks.
4. Generating Reports



## LEGAL CONSIDERATION

A software development agreement is a contract between a customer and a developer in which the customer hires the developer to build and deliver a certain piece of software. The following is a non-exhaustive list of five key topics that should be addressed in this type of contract.

**1. Scope of Work**

The most significant aspect of the agreement is the scope of work, which includes the project's objectives. It should include information on the deadline, tasks, deliverables, job quality, prices, and payment plan. It also establishes a project's requirements, such as the coding language utilized and the software's particular technical functions.

**2.Representations and Warranties**

The developer will make a number of promises to the customer about the product's performance as well as the quality and completeness of the documentation. Warranties are used to divide the risk between the developer and the customer.

**3.Ownership of Intellectual Property Rights**

Copyright often protects rights in software code and other work product developed under a software development agreement. Patents for software-implemented methods and devices, as well as trade secrets for the software's source code and related private documents and materials, may protect software rights.

**4.Open-Source Software**

Open-source software will almost always be used by the developer. Open-source software is computer software whose source code is made accessible to the public with a copyright holder's permission. It allows users to research, modify, and distribute software to anyone for any reason. The client may request a complete list of the developer's open source software. It's critical to get assurances from the developer that he or she has studied, comprehended, and adhered to the open-source software's specifications.

**5.Non-Infringement Indemnities**

There is a genuine possibility that the generated software will infringe on an existing patent due to the vast number of patents currently protecting software details and harsh penalty for patent infringement. Copyright infringement is also a possibility. Nonetheless, the risk of infringement, particularly patent infringement, makes the indemnity clause in a software development agreement particularly crucial.

## CONCLUSION

The system was designed in such a way that future modifications can be done easily. The following conclusions can be deduced from the development of the project.

* Automation of the entire system improves the efficiency
* It provides a friendly graphical user interface which proves to be much better.
* It gives appropriate access to the authorized users depending on their permissions.
* It effectively overcomes the delay in communications.
* Updating of information becomes so easier.
* System security, data security and reliability are the striking features.
* The System has adequate scope for modification in future if it is necessary.

## REFERANCES

Liv.ac.uk. (2021). *2CS21 - DESIGN AND STYLE*. [online] Available at: https://cgi.csc.liv.ac.uk/~frans/OldLectures/2CS21\_Ada/week9/designAndStyle.html [Accessed 2 Dec. 2021].

‌ Freetutes.com. (2021). *Degree of an Entity Relationship Type*. [online] Available at: https://www.freetutes.com/systemanalysis/sa7-degree-of-entity-relationship-type.html [Accessed 2 Dec. 2021].

Examcompetition.com. (2019). *Which of the following is a critical element of software quality assurance and represents the ultimate review of specification, design, and code generation?* [online] Available at: https://www.examcompetition.com/forum/170836/which-of-the-following-is-a-critical-element-of-so [Accessed 2 Dec. 2021].

Paperpass.net. (2021). *PROJECT RE\_PaperPass Report*. [online] Available at: https://view.paperpass.net/report/61a896ee303b4adp3/ [Accessed 2 Dec. 2021].

‌