

## **Project Report on ArtSpot**

### **Supervisor**

**Vaibhav Sir**  
**(Lecturer Computer)**

**Animesh Sahu**  
**(HOD)**

### **Submitted by**

- 1. Radheshyam Chaudhary**
- 2. Vikash Kumar Gupta**
- 3. Priyanshu Gautam**
- 4. Ajay Kumar**

# Index

1. Declaration
2. Acknowledgement
3. Introduction
4. Overview of Organization
5. Objectives
6. Existing System Description
7. Proposed System
8. Tools Used
9. System
10. Analysis
11. SDLC Phases
  - a. Feasibility Study
  - b. Report Approval
  - c. System Analysis
  - d. System Design
  - e. Coding
  - f. Testing
  - g. Implementation
  - h. Maintenance
12. Process Description
13. Project Model Used
14. ER- Diagram
15. Data Flow Diagram
16. Software Requirement
17. Specification
  - a. Hardware Requirement
  - b. Software Requirement
  - c. Support Maintenance
18. System Design
19. Low Level Design
20. Data Modeling
21. List of Tables
22. Structure of Tables
23. Testing
24. Input - Output
25. Future Scope
26. Conclusion

# DECLARATION

This is to certify that the project report entitled “ArtSpot” is done by me and my team is an authentic work carried out for the partial fulfillment of the requirements for the award of the Diploma in “Computer Science & Engineering”. The matter embodied in this project work has not been submitted earlier for award of any degree or B.Tech to the best of my knowledge and belief

Vikash Kumar Gupta  
Radhe Shyam Chaudhary  
Priyanshu Gautam  
Ajay Kumar

# **ACKNOWLEDGMENT**

We would like to express our sincere gratitude to everyone who contributed to the development and success of ArtSpot. This project would not have been possible without the dedication and hard work of our incredible team.

First and foremost, we would like to thank each team member for their unwavering commitment, creativity, and collaboration throughout this journey. Your unique talents and perspectives have been invaluable in bringing ArtSpot to life.

We extend our heartfelt appreciation to our mentors and advisors, whose guidance and support have been instrumental in shaping the direction of this project. Your insights and encouragement have helped us overcome challenges and achieve our goals.

We would also like to thank our friends and family for their constant support and understanding, providing us with the motivation and encouragement needed to see this project through to completion.

Lastly, we are grateful to our community and all the artists who inspired ArtSpot. Your passion for art and creativity has been a constant source of inspiration, driving us to create a platform that celebrates and showcases artistic talent.

Thank you all for being a part of this incredible journey. We look forward to continuing to build and grow ArtSpot with your ongoing support and collaboration.

With gratitude,

**The ArtSpot Team**

## **Introduction**

ArtSpot portal helps that one can take any information about that particular art for which project is concerned. ArtSpot is a collection of art you can buy and download. ArtSpot can store in dataset , help of artist, home page , art gallery, User profile, login page and upload art.

## **Overview of Organization**

Government Polytechnic Unnao Engineering College is a premier institution in Uttar Pradesh, committed to providing high-quality technical education and fostering innovation and research in the field of engineering. Established with the aim of nurturing skilled professionals, the college has become a hub for aspiring engineers and technocrats.

# Software Requirement

## Software Requirements:

### 1. Development Environment:

- Visual Studio
- .NET Framework/.NET Core

### 2. Database Management:

- Microsoft SQL Server

### 3. Version Control:

- Git

### 4. Frontend Technologies:

- HTML, CSS, JavaScript
- jQuery
- Bootstrap

### 5. Additional Tools and Libraries:

- NuGet
- AutoMapper
- Entity Framework
- TensorFlow
- scikit-learn
- Keras

### 6. AI Services and APIs:

- Microsoft Azure Cognitive Services
- Google Cloud AI Platform
- Amazon Web Services (AWS) AI Services

### 7. Development Tools and Environments:

- Visual Studio Code

### 8. Integration and Deployment:

- APIs and SDKs
- Deployment Strategies

# OBJECTIVES

The objective of ArtSpot is to create a vibrant online platform that connects artists, art enthusiasts, and collectors worldwide. We aim to:

1. **Promote Artistic Expression:** Provide a space for artists to showcase their work and explore diverse styles.
2. **Enhance Visibility:** Increase artists' reach through digital marketing and social media.
3. **Foster Community Engagement:** Build a supportive community for collaboration and idea exchange.
4. **Facilitate Art Transactions:** Offer a secure platform for buying, selling, and commissioning artworks.
5. **Educate and Inspire:** Provide educational content to inspire and inform the art community.
6. **Leverage Technology:** Use innovative tech for immersive user experiences like virtual galleries and AR exhibitions.
7. **Support Sustainability:** Encourage environmentally sustainable practices and social impact initiatives.

# PROBLEM DEFINITION

In this section we shall discuss the limitation and drawback of the existing system that forced us to take up this project. Really that work was very typical to manage the daily errors free records and adding or removing any node from server. This problem produces a need to change the existing system. Some of these shortcomings are being discussed below: - ☒

## **Low Functionality**

With the existing system, the biggest problem was the low functionality. The problem faced hampered the work. For small task like adding any new node to server or deleting a node or keeping daily record we have to appoint minimum two or three employee. ☒

## **Erroneous Input and Output**

In the existing system, humans performed all the tasks. As in the human tendency, error is also a possibility. Therefore, the inputs entered by the person who is working in the Company, in the registers may not be absolutely foolproof and may be erroneous. As a result of wrong input, the output reports etc.will also be wrong which would in turn affect the performance.

## **Portability Problem**

System that existed previously was manual. As a result, the system was less portable. One has to carry the loads of many registers to take the data from one place to another. A big problem was that the system was less flexible and if we wanted to calculate yearly or monthly maintenance report or efficiency report, then it was a big headache. ☒

## **Security**

Security concerns were also one of the motives of the Company for the need of software. In the registers, the data is not secure as anybody can tamper with the data written in the registers. While in this software, just a password makes it absolutely secure from the reach of unauthorized persons. ☒

## **Data Redundancy**

In the case of manual system, the registers are maintained in which, a lot of data is written.

## **Complexity in Work**

In manual system whenever a record is to be updated or to be deleted a lot of cutting and overwriting needs to be done on the registers that are concerned that are deleted or updated record, which makes the work very complex.



# SYSTEM ANALYSIS

## **Phases:**

System Development Life Cycle (SDLC) mainly consists of the following 7 phases which can be detailed:-

### **Preliminary Investigation:**

This is the first phase of the system development life cycle. In this phase we tend to find out the needs of the client —what exactly does the client want? Before the development of any system the important point is to know the needs, objectives and scope of the system.

### **Feasibility Study:**

Feasibility study is these conducted step of the system development life cycle. Things are always easy at the beginning in any software process. In fact nothing is infeasible with unlimited time and resources. But it is not the fact. So, practically we have to do in limited resources in a restricted time margin. So for the system to be feasible, following points we have to consider. The feasibility study is conducted to check whether the candidate system is feasible. The system which is selected to be the best against the criteria is thereafter designed and developed. The feasibility study takes in to consideration, the risks involved in the project development beforehand. Therefore in this phase we have to do feasibility study which is the test of the website according to its workability, impact on the organization, ability to meet user need and effective use of resources. We do the feasibility study for website to analyze the risks, costs and benefits relating to economics, technology and user organization. There are several types of feasibility depending on the aspect they cover. Import of these includes:

### ☒ **Technical Feasibility:**

This is an important outcome of preliminary investigation .It comprise of following questions:- ☒  
Can the work of project be done with the current equipment ,existing software and available man power resource?

If Technology is required what are the possibilities that it can be developed?

**Economic Feasibility:** It deals with question related to the economy .It comprise of the following questions:-

Are there sufficient benefits in creating the system to make the cost acceptable? ☒  
Are the costs of not creating the system so great that the project must be undertaken?

**legal Feasibility:**☒ It deals with the question related to the legal issues. It comprise of the following questions:-

☒ Contract Signing ☒ Software License agreement ☒ Issues related to cyber laws. ☒  
Legal issues relating to the man power contract.

☒ Operational Feasibility:☒ The operational feasibility consists of the following activity:- ☒

Will the system be useful if it is developed &implemented? ☒

Will the reb resistance from employee?

### ☒ **Social &Behavioral Feasibility:**

It deals with the various issues related to the human behavior like:-

☒ Whether the user be able to adapt a new change or not? ☒ Whether the ambiance we are providing suits the user or not?

**Request Approval:-** Request approval is the third phase of system development lifecycle. Request approval is the phase in which all the requirements which would be provide in the system are stated. The request approval is a sort of agreement between the client and the company which is building his software. Both the parties should be mutually agreed on the stated requirements.

## **System Analysis:**

System analysis is the phase following the phase of the quest approval. In this phase we end to analyze the overall system which we have to build. System analysis is the crucial part in SDLC.

## **System Design:**

System design means the designing of the system. The System can be done in either of the following two ways:-

1. Logical System Design
2. Physical System Design

## **Coding:-**

Coding is the phase in which a developer codes using any programming languages. Coding constitutes only 20 % of the whole project and which is easier to write. The coding work is also done in the teams; development of the system is usually done under the modular programming style, which can be either top-down approach or bottom-up approach.

## **Testing:-**

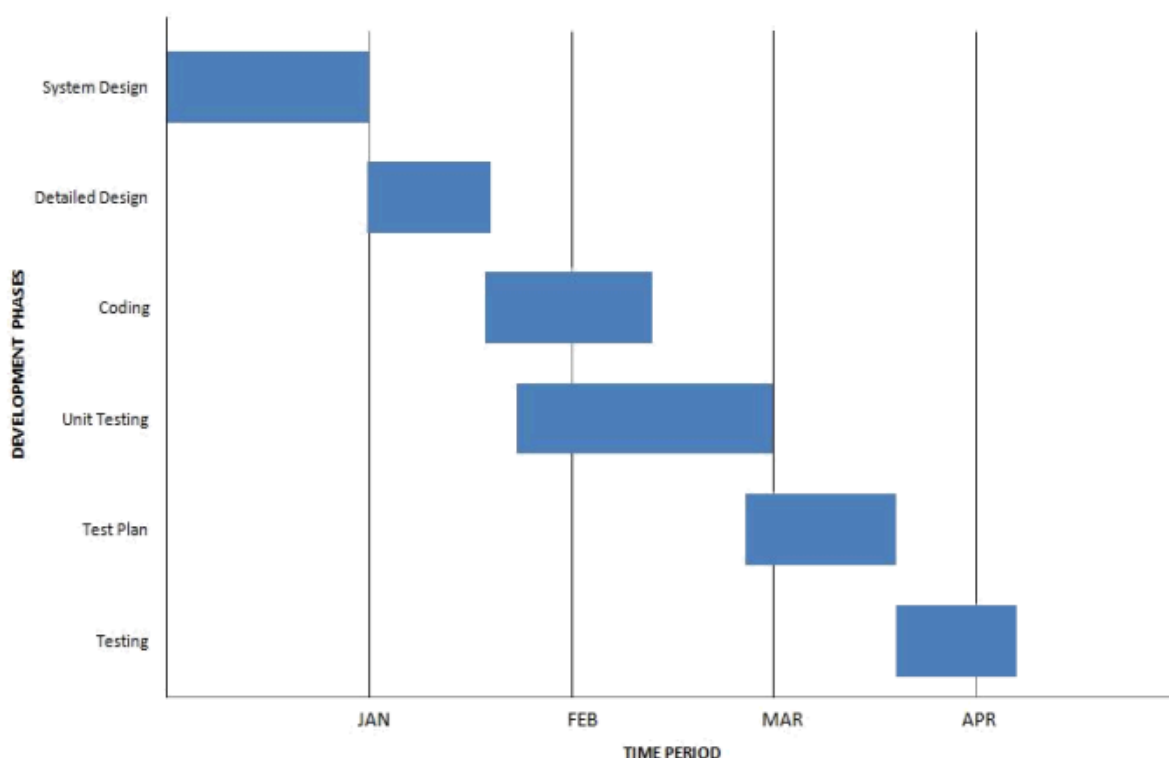
Testing is the phase in which the system that has been developed is tested. Testing comprises of the 60% of the overall development of the system. Testing of the system is important because testing aims to uncover the different errors in the system. There are various different testing techniques that can be used for the testing of the system.

## **Maintenance: -**

Merely developing the system is not important but also maintenance is important. The company that has built the system provides for some time free of cost maintenance to the client and after that period it is usually a paid service.

## Process Description

Gantt charts mainly used to allocate resources to activities. The resources allocated to activities include staff, hardware, and software. Gantt charts (named after its developer Henry Gantt) are useful for resource planning. A Gantt chart is special type of bar chart where each bar represents an activity. The bars are drawn along a timeline. The length of each bar is proportional to the duration of the time planned for the corresponding activity. Gantt chart is a project scheduling technique. Progress can be represented easily in a Gantt chart, by coloring each milestone when completed. The project will start in the month of January and end after 4 months at the beginning of April.



# PROJECT MODEL USED

## Iterative Enhancement Model

This model has the same phases as the waterfall model, but with fewer restrictions. Generally the phases occur in the same order as in the waterfall model, but they may be conducted in several cycles.

Useable product is released at the end of the each cycle, with each release providing additional functionality. Customers and developers specify as many requirements as possible and prepare a SRS document. Developers and customers then prioritize these requirements. Developers implement the specified requirements in one or more cycles of design, implementation and test based on the defined priority.

The procedure itself consists of the initialization step, the iteration step, and the Project Control List. The initialization step creates a base version of the system. The goal for this initial

implementation is to create a product to which the user can react. It should offer a sampling of the key aspects of the problem and provide a solution that is simple enough to understand and implement easily. To guide the iteration process, a project control list is created that contains record of all tasks that need to be performed. It includes such items as new features to be implemented and areas of redesign of the existing solution. The control list is constantly being revised as a result of the analysis phase.

The iteration involves the redesign and implementation of iteration is to be simple, straightforward, and modular, supporting redesign at that stage or as a task added to the project

control list. The level of design detail is not dictated by the iterative approach. In a lightweight

iterative project the code may represent the major source of documentation of the system; however, in a critical iterative project a formal Software Design Document may be used. The analysis of an iteration is based upon user feedback, and the program analysis facilities available.

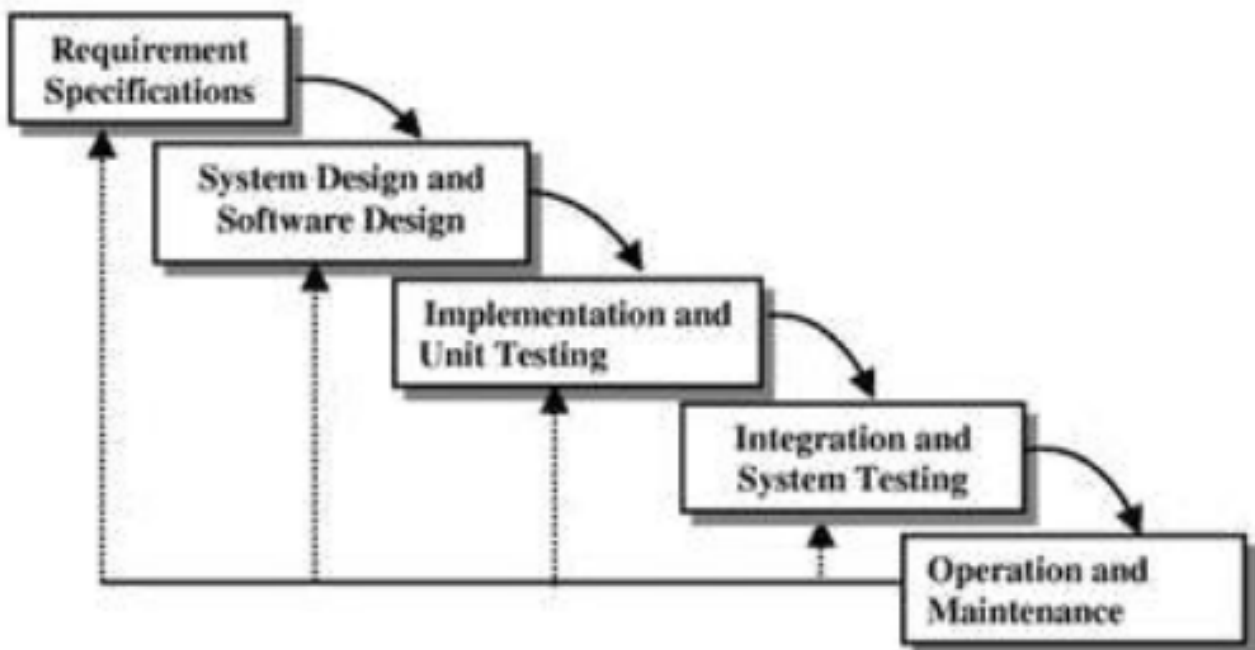
It involves analysis of the structure, modularity, usability, reliability, efficiency, & achievement

of goals. The project control list is modified in light of the analysis results.

## PHASES:

Incremental development slices the system functionality into increments (portions). In each increment, a slice of functionality is delivered through cross-discipline work, from the requirements to the deployment. The unified process groups increments/iterations into phases: inception, elaboration, construction, and transition.

- ☒ Inception identifies project scope, requirements (functional and non-functional) and risks at a high level but in enough detail that work can be estimated.
- ☒ Elaboration delivers a working architecture that mitigates the top risks and fulfills the nonfunctional requirements. ☒
- Construction incrementally fills-in the architecture with production-ready code produced from analysis, design, implementation, and testing of the functional requirements. ☒
- Transition delivers the system into the production operating environment.



# ER-Diagram



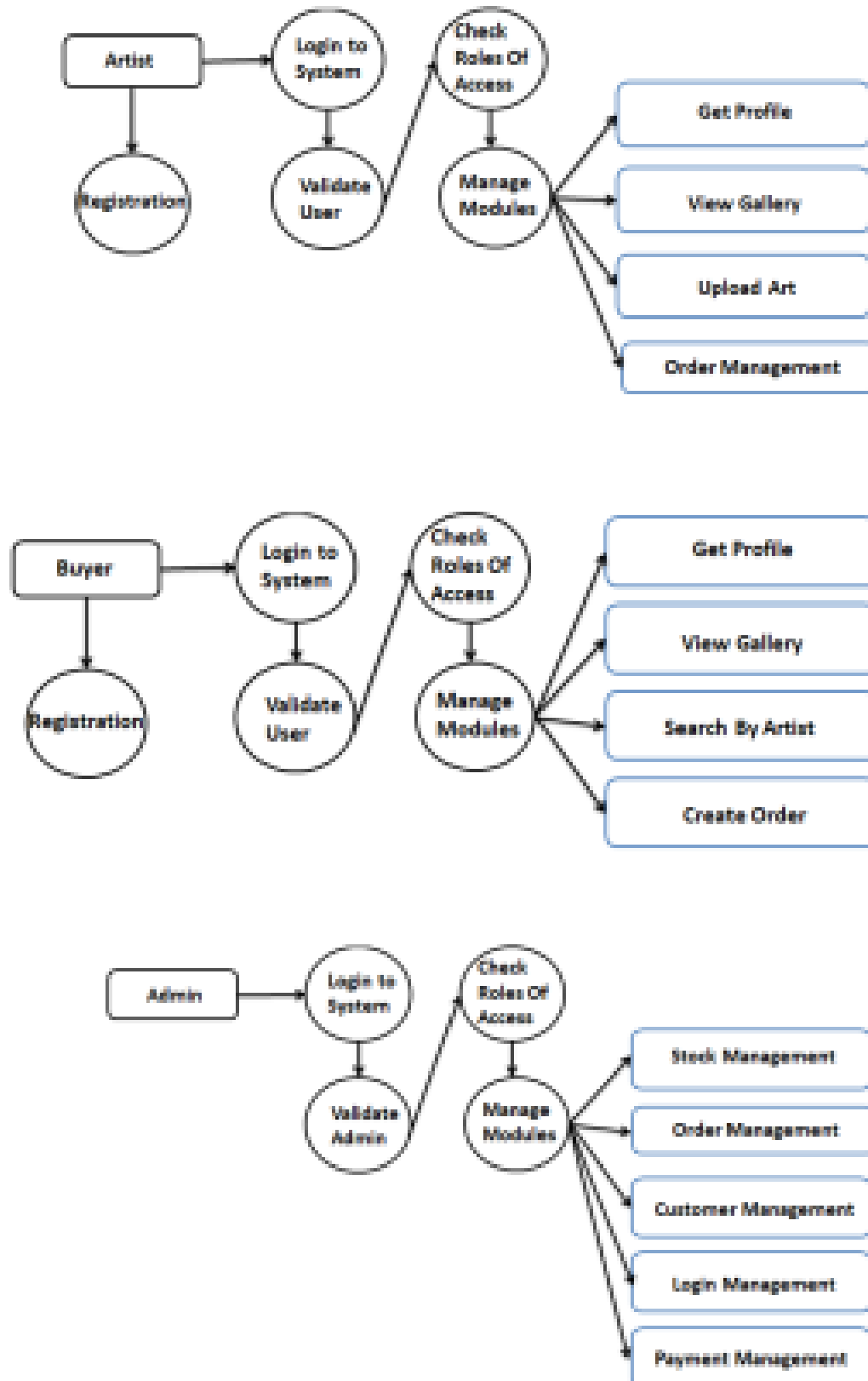
## Introduction:-

In software engineering, an entity-relationship model(ERM) is an abstract and conceptual representation of data. Entity-relationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion. Diagrams created by this process are called entity-relationship diagrams, ER diagrams, or ERDs. ER Diagrams depicts relationship between data objects. The attribute of each data objects noted in the entity-relationship diagram can be described using a data object description. Entity relationship diagram is very basic, conceptual model of data and it is fundamental to the physical database design. This analysis is then used to organize data as relations, normalizing relations, and obtaining a Relational database.

The entity-relationship model for data uses three features to describe data. These are:

1. Entities which specify distinct real-world items in an application.
2. Relationship, which connect entities and represent meaningful dependencies between them.
3. Attributes which specify properties of entities & relationships.

## Data Flow Diagram





## **SOFTWARE REQUIREMENT SPECIFICATION**

A requirements specification for a software system is a complete description of the behavior of a system to be developed and it includes a set of use cases that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements.

Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance engineering requirements, quality standards, or design constraints). Requirements are a sub-field of software engineering that deals with the elicitation, analysis, specification, and validation of requirements for software.

The software requirement specification document enlists all necessary requirements for project development. To derive the requirements we need to have clear and thorough understanding of the products to be developed. This is prepared after detailed communications with project team and the customer.

### **Hardware Requirements**

#### ***(A) For Client Side –***

- Internet explorer 8, Chrome or any supported browser
- Pentium 4
- 256 MB RAM
- 1 GB HDD

#### ***(B) For Server Side –***

- Xeon based or higher end Server
- 4 GB RAM
- 200 GB disk space

### **Software Requirements**

- Client on Internet: Web Browser, Operating System (any).
- Client on Intranet: Client Software, Web Browser, Operating System (any).
- Web Server: IIS 7 or higher, Operating System (windows server).
- Application framework: Microsoft .NET Framework 4.0.
- Data Base Server: MS SQL SERVER 2008 R2, Operating System (windows Server).
- Development End: Visual Studio 2012 (ASP.NET, HTML, JavaScript), MS SQL SERVER 2008, OS (Windows Server), Web Server (IIS 6 or higher).
- Management studio 2012

# SUPPORT AND MAINTENANCE

One year free support for rectifying system bugs including front end and back end will be provided. During warranty period Software Engineers will be responsible for removing bugs and improving it. After one year support can be extended @ 20% of the total product deployment cost

## SYSTEM DESIGN APPROACH

### *Top – Down designing:*

The top - down designing approach started with major components of the system. It is a stepwise refinement which starts from an abstract design, in each steps the design is refined two or more concrete levels until we reach a level where no – more refinement is possible or not needed.



### *Bottom – Up designing:*

In bottom – up designing the most basic and primitive components are designed first, and we proceed to higher level components. We work with layers of abstractions and abstraction are implemented until the stage is reached where the operations supported by the layer is complete.



# DATA MODELING

## LIST OF TABLES:

1. Tbl\_Registration
2. Tbl\_Enquiry
3. Tbl\_Notification
4. Tbl\_Login

### 1. Tbl Registration:-

Field Name	Data Type	Status
FirstName	Varchar(100)	Not null
LastName	Varchar(7)	Not null
Gender	Varchar(100)	Not null(Primary key)
DOB	Varchar(20)	Not null
Address	Varchar(Max)	Not null
Contact	Varchar(Max)	Not null
Email Id	Varchar(20)	Not null
Password	Varchar(30)	Not null
Profile	Varchar(70)	Not null

### 2. Tbl\_Enquiry

Field Name	Data Type	Status
Id	Int	Not null (primary key)
Name	varchar(50)	Not null
Mobile	Varchar(30)	Not null
EmailId	Varchar(90)	Not null
Message	Varchar(100)	Not null
Enq Date	Datetime	Null

### 3. Tbl Notification

Field Name	Data Type	Status
ID	Int	Not null(primary key)
Addnotification	Varchar(100)	Not null
Notidate	Datetime	Null

### 4. Tbl Login

Field Name	Data Type	Status
ID	Int	Not null(primary key)
EmailId	Varchar(100)	Not null
Password	Varchar(60)	Not null

## Low Level Design

**Description:** Low Level Design creation is one of the most important activities in the development of any software product. The low level design document gives the design of the actual software application. Low level design document is based on High Level Design document . It defines internal logic of every sub module. A good low level design document will make the application very easy to develop by the developer. An effective design document results in very low efforts in developing a Software product.

Each project's low level design document should provide a complete and detailed specification of the design for the software that will be developed in the project, including the classes, member and non-member functions, and associations between classes that are involved. The low level design document should contain a listing of the declarations of all the classes, non-member-functions, and class member functions that will be defined during the subsequent implementation stage, along with the associations between those classes and any other details of those classes (such as member variables) that are firmly determined by the low level design stage. The low level design document should also describe the classes, function signatures, associations, and any other appropriate details, which will be involved in testing and evaluating the project according to the evaluation plan defined in the project's requirements document

## TESTING

Testing is the integral part of any System Development Life Cycle insufficient and interested application tends to crash and result in loss of economic and manpower investment besides user's dissatisfaction and downfall of reputation. "Software Testing can be looked upon as one among much process, an organization performs, and that provides the last opportunity to correct any flaws in the developed system. Software Testing includes selecting test data that have more probability of giving errors." The first step in System testing is to develop the plan that all aspect of system .Complements, Correctness, Reliability and Maintainability.

# Type of Testing

## **Black Box (Functional) Testing:**

Testing against specification of system or component. Study it by examining its inputs and related outputs. Key is to devise inputs that have a higher likelihood of causing outputs that reveal the presence of defects. Use experience and knowledge of domain to identify such test cases. Failing this a systematic approach may be necessary. Equivalence partitioning is where the input to a program falls into a number of classes, e.g. positive numbers vs. negative numbers. Programs normally behave the same way for each member of a class. Partitions exist for both input and output. Partitions may be discrete or overlap. Invalid data (i.e. outside the normal partitions) is one or more partitions that should be tested.

Internal System design is not considered in this type of testing. Tests are based on requirements and functionality. This type of test case design method focuses on the functional requirements of the software, ignoring the control structure of the program. Black box testing attempts to find errors in the following categories:

- Incorrect or missing functions. ☒
- Interface errors.
- Errors in data structures or external database access. ☒
- Performance errors. ☒
- Initialization and termination errors

## **White Box Testing**

Testing based on knowledge of structure of component (e.g. by looking at source code). Advantage is that structure of code can be used to find out how many test case need to be performed. Knowledge of the algorithm (examination of the code) can be used to identify the equivalence partitions. Path testing is where the tester aims to exercise every independent execution path through the component. All conditional statements tested for both true and false cases. If a unit has  $n$  control statements, there will be up to  $2^n$  possible paths through it. This demonstrates that it is much easier to test small program units than large ones. Flow graphs are a pictorial representation of the paths of control through a program (ignoring assignments, procedure calls and I/O statements).

**Unit Testing:**

Unit testing concentrates on each unit of the software as implemented in the code. This is done to check syntax and logical errors in programs. At this stage, the test focuses on each module individually, assuring that it functions properly as a unit. In our case, we used extensive whitebox testing at the unit testing stage. A developer and his team typically do the unit testing do the unit testing is done in parallel with coding; it includes testing each function and procedure.

**Incremental Integration Testing:**

Bottom up approach for testing i.e. continuous testing of an application as new functionality is added; Application functionality and modules should be independent enough to test separately done by programmers or by testers.

**Integration Testing:**

Testing of integration modules to verify combined functionality after integration .Modules are typically code modules, individual applications, client and server and distributed systems.

**Functional Testing:**

This type of testing ignores the internal parts and focus on the output is as per requirement or not .Black box type testing geared to functionality requirements of an application.

**System Testing:**

Entire system is tested as per the requirements. Black box type test that is based on overall requirement specifications covers all combined parts of a system.

**Beta Testing:**

Testing typically done by end-users or others. This is final testing before releasing application for commercial purpose.

# Description of DB Manager

```
using ArtSpot.Models;
using Microsoft.Ajax.Utilities;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Data.Entity;

namespace ArtSpot.Controllers
{
    0 references
    public class HomeController : Controller
    {
        .....artspotEntities db = new artspotEntities();

        .....//public ActionResult Index()
        .....//{
        .....//var tbl_product = db.tbl_product.Include(t => t.tbl_category).Include(t => t.tbl_user);
        .....//return View(tbl_product.ToList());
        .....//}
        0 references
        public ActionResult Index(string searchBy, string search)
        {
            .....IQueryable<tbl_product> products = db.tbl_product.Include(t => t.tbl_category).Include(t => t.tbl_user);

            .....if (!string.IsNullOrEmpty(search))
            .....{
            .....if (searchBy == "name")
            .....{
            .....products = products.Where(p => p.pro_name.StartsWith(search));
            .....}
            .....else if (searchBy == "category")
            .....{
            .....products = products.Where(p => p.tbl_category.cat_name.StartsWith(search));
            .....}
            .....}

            .....return View(products.ToList());
            .....}

        0 references
        public ActionResult About()
        {
            .....ViewBag.Message = "Your application description page.";

            .....return View();
            .....}

        0 references
        public ActionResult Contact()
        {
            .....ViewBag.Message = "Your contact page.";

            .....return View();
            .....}

        0 references
        public ActionResult User_Login()
        {
            .....return View();
            .....}
    }
}
```

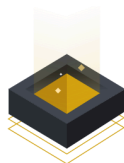
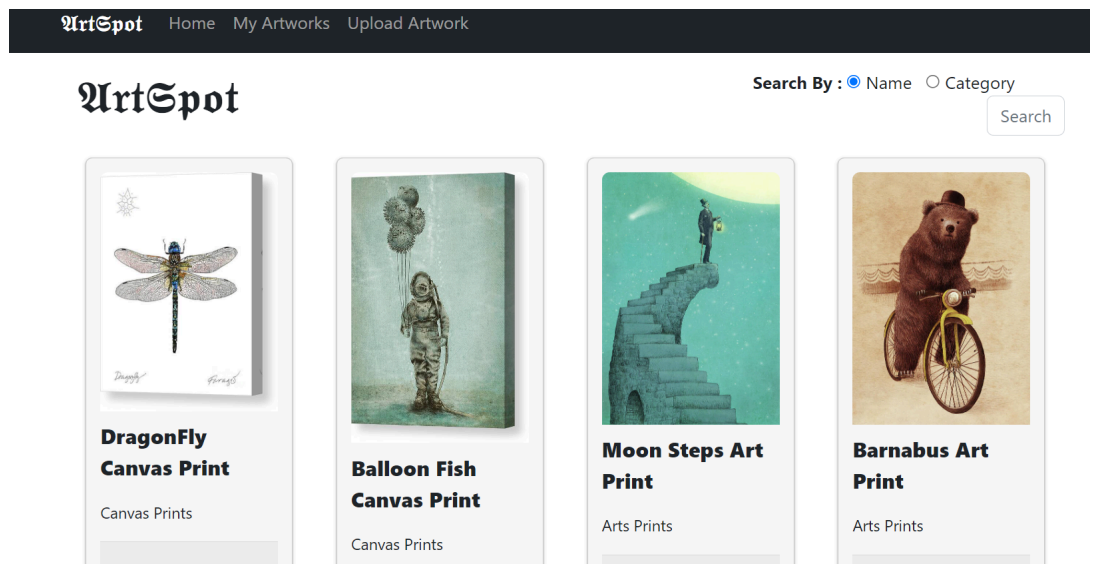
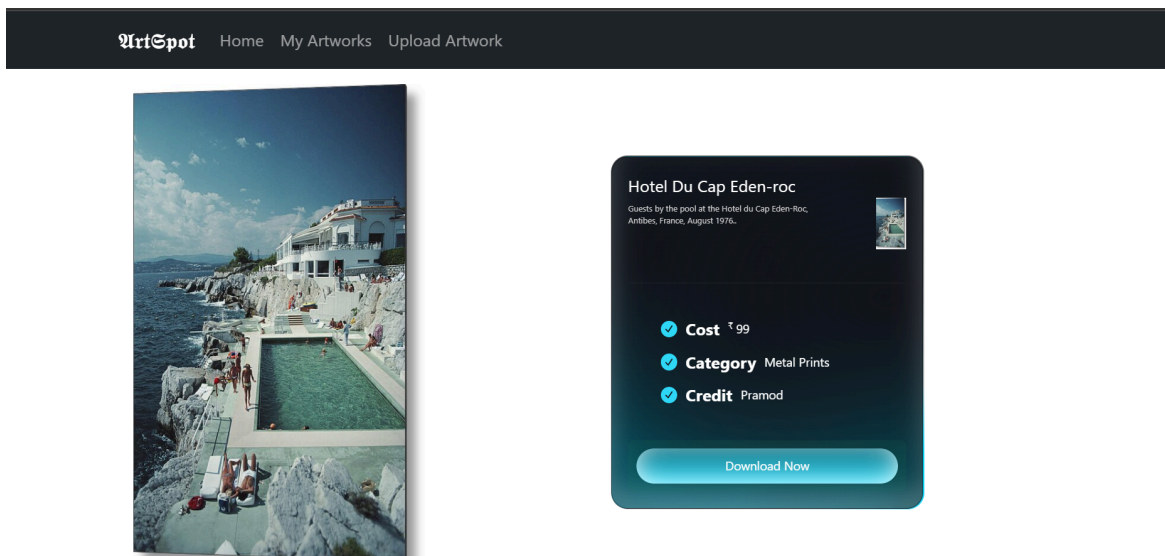
```

namespace ArtSpot.Models
{
    ...using System;
    ...using System.Data.Entity;
    ...using System.Data.Entity.Infrastructure;
    ...
    11 references
    ...public partial class artspotEntities : DbContext
    ...{
        5 references
        ...public artspotEntities()
        ...{
        ...: base("name=artspotEntities")
        ...{
        ...}
        ...
        0 references
        ...protected override void OnModelCreating(DbModelBuilder modelBuilder)
        ...{
        ...throw new UnintentionalCodeFirstException();
        ...}
        ...
        5 references
        ...public virtual DbSet<tbl_admin> tbl_admin { get; set; }
        11 references
        ...public virtual DbSet<tbl_category> tbl_category { get; set; }
        10 references
        ...public virtual DbSet<tbl_product> tbl_product { get; set; }
        12 references
        ...public virtual DbSet<tbl_user> tbl_user { get; set; }
        ...}
}

```



# User Screenshots



## About Us

### Welcome to ArtSpot!

At ArtSpot, we believe in the transformative power of art and its ability to bring people together, inspire creativity, and enrich lives. Our platform is dedicated to creating a vibrant online community where artists and art enthusiasts can connect, share, and discover incredible artworks from around the world.

### Our Mission

Our mission is to provide a dynamic and inclusive space for artists to showcase their work and for art lovers to explore and purchase unique pieces. We aim to democratize the art world by making it accessible to everyone, whether you're an emerging artist looking to gain exposure or a seasoned collector searching for the next addition to your gallery.








### What We Offer

# Admin Dashboard



## Index

[Create New](#)

u_image	u_name	u_email	u_password	u_contact	
	Vikash	vikash@gmail.com	123	8226935738	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
	Ajay	ajay@gmail.com	123	9586316872	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
	Radheshyam	radhe@gmail.com	123	6354982678	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
	Akash	akash@gmail.com	123	5486971583	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
	Sandeep	sandeep@gmail.com	123	6279826459	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
	Kushal	kushal@gmail.com	123	63587329814	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>
	Pramod	pramod@gmail.com	123	8294638726	<a href="#">Edit</a>   <a href="#">Details</a>   <a href="#">Delete</a>

```
Model1.edmx [Diagram1]    About.cshtml    _UserLayoutPage.cshtml    Layout.cshtml    Admin_Dashboard.cshtml    Pro_Details.cshtml    Details.cshtml    All_Art.cshtml    Index.cshtml

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8" />
5 <meta name="viewport" content="width=device-width, initial-scale=1.0">
6 <title>@ViewBag.Title - ArtSpot</title>
7 @Styles.Render("~/Content/css")
8 @Scripts.Render("~/bundles/modernizr")
9 </head>
10 <body>
11 <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-dark bg-dark">
12 <div class="container">
13 <div>@Html.ActionLink("ArtSpot", "Index", "Home", new { area = "" }, new { @class = "navbar-brand" })</div>
14 <div>@Html.ActionLink("Index", "Index", "Home", new { area = "" }, new { @class = "nav-link" })</div>
15 <div>@Html.ActionLink("About Us", "About", "Home", new { area = "" }, new { @class = "nav-link" })</div>
16 <div>@Html.ActionLink("Contact Us", "Contact", "Home", new { area = "" }, new { @class = "nav-link" })</div>
17 <div>@Html.ActionLink("Login", "User_Login", "Home", new { area = "" }, new { @class = "nav-link" })</div>
18 <div>@Html.ActionLink("Register", "Create", "User", new { area = "" }, new { @class = "nav-link" })</div>
19 </div>
20 </nav>
21 <div class="container body-content">
22 @RenderBody()
23 <br />
24 </div>
25 <hr />
26 <div class="container-fluid bg-dark m-0 p-0">
27 <div class="row p-0 m-0 text-center" style="text-decoration:none; color:white">
28 <div class="col-sm-4">
29 <ul style="text-decoration:none; list-style-type:none">
30 <li>@Html.ActionLink("Home", "Index", "Home", new { area = "" }, new { @class = "nav-link" })</li>
31 <li>@Html.ActionLink("About Us", "About", "Home", new { area = "" }, new { @class = "nav-link" })</li>
32 <li>@Html.ActionLink("Contact Us", "Contact", "Home", new { area = "" }, new { @class = "nav-link" })</li>
33 <li>@Html.ActionLink("Login", "User_Login", "Home", new { area = "" }, new { @class = "nav-link" })</li>
34 <li>@Html.ActionLink("Register", "Create", "User", new { area = "" }, new { @class = "nav-link" })</li>
35 </ul>
36 </div>
37 </div>
38 </div>
39 </div>
40 </div>
41 </div>
42 </div>
43 </div>
44 </div>
45 </div>
46 </div>
47 </div>
48 </div>
49 </div>
50 </div>
51 </div>
52 </div>
53 </div>
```

```
Model1.edmx [Diagram1]    About.cshtml    _UserLayoutPage.cshtml    Layout.cshtml    Admin_Dashboard.cshtml    Pro_Details.cshtml    Details.cshtml    All_Art.cshtml    Index.cshtml

37 <div class="container-fluid bg-dark m-0 p-0">
38 <div class="row p-0 m-0 text-center" style="text-decoration:none; color:white">
39 <div class="col-sm-4">
40 <ul style="text-decoration:none; list-style-type:none">
41 <li>@Html.ActionLink("Home", "Index", "Home", new { area = "" }, new { @class = "nav-link" })</li>
42 <li>@Html.ActionLink("About Us", "About", "Home", new { area = "" }, new { @class = "nav-link" })</li>
43 <li>@Html.ActionLink("Contact Us", "Contact", "Home", new { area = "" }, new { @class = "nav-link" })</li>
44 <li>@Html.ActionLink("Login", "User_Login", "Home", new { area = "" }, new { @class = "nav-link" })</li>
45 <li>@Html.ActionLink("Register", "Create", "User", new { area = "" }, new { @class = "nav-link" })</li>
46 </ul>
47 </div>
48 <div class="col-sm-4">
49 <ul style="text-decoration:none; list-style-type:none">
50 <li>@Html.ActionLink("Sign Up", "Create", "User", new { area = "" }, new { @class = "nav-link" })</li>
51 <li>@Html.ActionLink("Login", "User_Login", "Home", new { area = "" }, new { @class = "nav-link" })</li>
52 </ul>
53 </div>
54 </div>
55 </div>
56 </div>
57 </div>
58 </div>
59 </div>
60 </div>
61 </div>
62 </div>
63 </div>
64 </div>
65 </div>
66 </div>
67 </div>
68 </div>
69 </div>
70 </div>
71 </div>
72 </div>
73 </div>
74 </div>
75 </div>
76 </div>
77 </div>
78 </div>
79 </div>
80 </div>
81 </div>
82 </div>
83 </div>
84 </div>
85 </div>
86 </div>
87 </div>
88 </div>
89 </div>
90 </div>
```

## **Future Scope**

Following modification or upgrades can be done in system.

1. Our Future scope and plannings as well as NGOs working for that village in various areas like education, health, women empowerment etc. The current system in use is a paper-based system.
2. Any villager can get information about his KHASRA number and all other status through the E- village portal

## **CONCLUSION**

ArtSpot stands as a dynamic and innovative platform that bridges the gap between artists and art enthusiasts worldwide. By providing a space for artistic expression, community engagement, and art transactions, ArtSpot empowers artists to showcase their talent and reach a broader audience. Our commitment to leveraging technology and promoting sustainability ensures that ArtSpot is not only a marketplace but also a thriving community dedicated to the appreciation and advancement of art.