

SDLC REPORT ON “TRIP ENGINEERS”

Submitted by :

Vikas

UGI(Prayagraj)

INTRODUCTION

Over the years, tourism has continued to gain massive interest at a global scale. It is a major foreign exchange earner for a good number of advanced and emerging economies. It is also true that information explosion makes it cumbersome times to access relevant information to enhance decision making. This has given rise to the emergence of intelligent systems or mechanisms that facilitate quick access to relevant content found in the Internet. For developing countries, tourism is one of the untapped but potentially big income generator. There are about 142 tourist destinations that spread across the world. Whereas some exist naturally, others are man made . In this era that has witnessed rapid advances in information technology, information overload has become a serious problem to those seeking for information online. Recently, intelligent search mechanisms have been deployed on the web that shows that the problem of information overload can be partially eliminated by providing a platform with more intelligence to assist tourists in the search for relevant information. Google.com is an example of an intelligent search engine that helps users with information and another class of intelligent system that has proven relevant in addressing the problem of information overload are recommender systems .In the aspect of tourism, Internet and web technologies have made more readily available information on tourist locations, accommodations, transportation, shopping, food, festivals, and other attractions, thus improving tourism experience .The goal of this research is to design and implement intelligent platform that will aid tourists to have access to information on tourist locations thus help fasten their decision making process.

SOFTWARE AND HARDWARE **REQUIREMENT**

1. Operating System (32 or 64bit Windows 7/vista/XP and latter , Linux, mac OS, android OS)
2. Processor (1 dual core or single core processor)
3. Internet browser (Mozilla Firefox (most suitable), opera mini, Google chrome, or internet explorer)

The hardware requirements are:

1. CPU (Pentium III, 950MHz, CPU)
2. Memory (256MB RAM)
3. Video graphics adaptor (16bit VGA)
4. Network card (1GB Ethernet)
5. Hard disk (5GB)

Languages used are:

HTML
CSS
JAVASCRIPT

Tools Used are:

VS Code

SDLC Phases

The entire SDLC process divided into the following SDLC steps:

- Phase 1: Requirement collection and analysis
- Phase 2: Feasibility study
- Phase 3: Design
- Phase 4: Testing
- Phase 5: Maintenance

Phase 1: Requirement collection and analysis

The requirement is the first stage in the SDLC process. It is conducted by the senior team members with inputs from all the stakeholders and domain experts in the industry. Planning for the quality assurance requirements and recognition of the risks involved is also done at this stage.

This stage gives a clearer picture of the scope of the entire project and the anticipated issues, opportunities, and directives which triggered the project.

Requirements Gathering stage need teams to get detailed and precise requirements. This helps company to finalize the necessary timeline to finish the work of that system.

Phase 2: Feasibility study

Once the requirement analysis phase is completed the next sdlc step is to define and document software needs. This process conducted with the help of 'Software Requirement Specification' document also known as 'SRS' document. It includes everything which should be designed and developed during the project life cycle.

There are mainly five types of feasibilities checks:

- **Economic:** Can we complete the project within the budget or not?
- **Legal:** Can we handle this project as cyber law and other regulatory framework/compliances.
- **Operation feasibility:** Can we create operations which is expected by the client?
- **Technical:** Need to check whether the current computer system can support the software
- **Schedule:** Decide that the project can be completed within the given schedule or not.

Phase 3: Design

In this third phase, the system and software design documents are prepared as per the requirement specification document. This helps define overall system architecture.

This design phase serves as input for the next phase of the model.

There are two kinds of design documents developed in this phase:

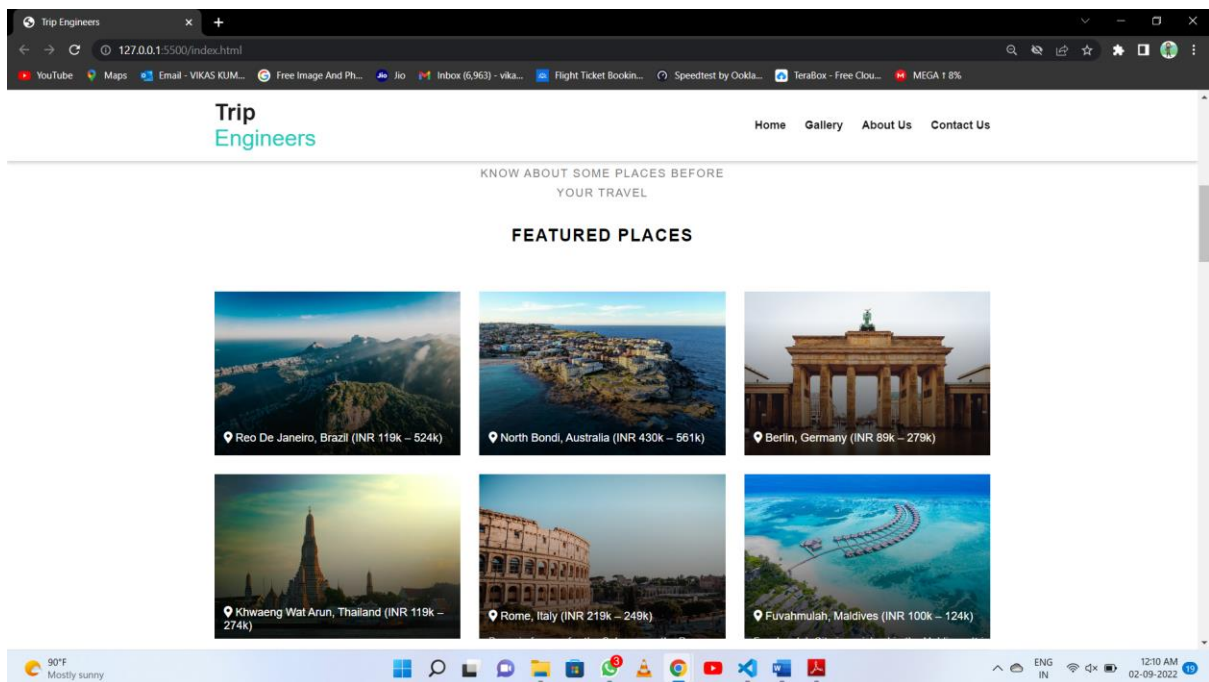
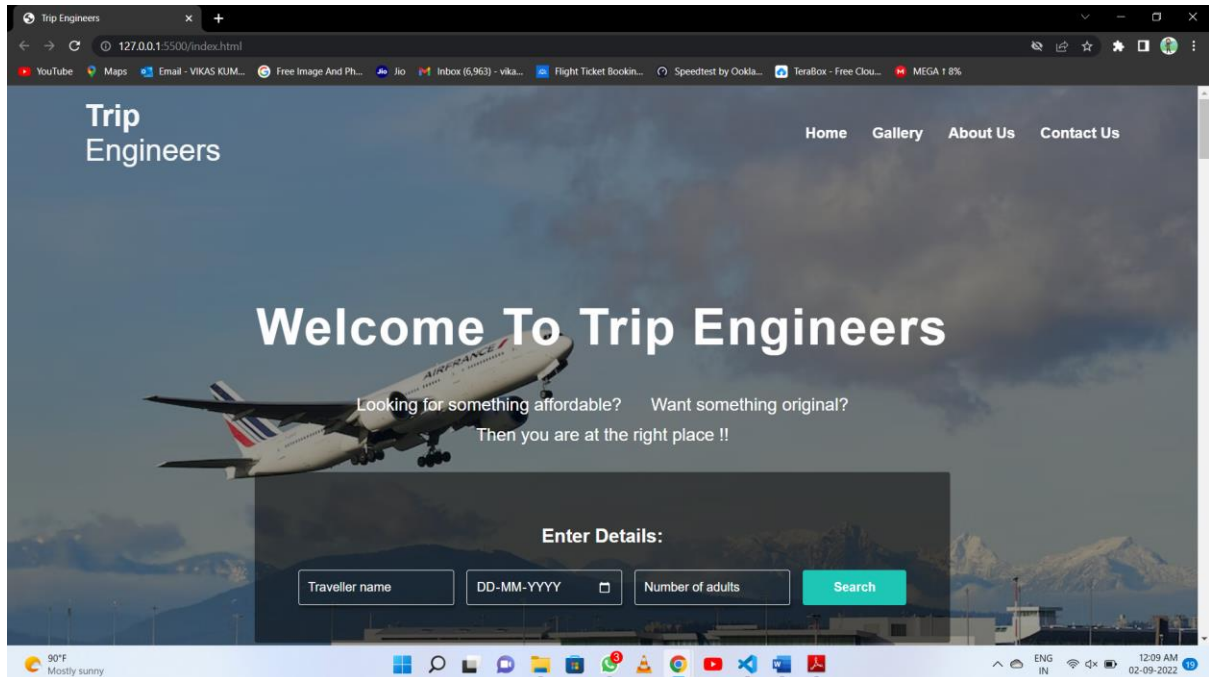
High-Level Design (HLD)

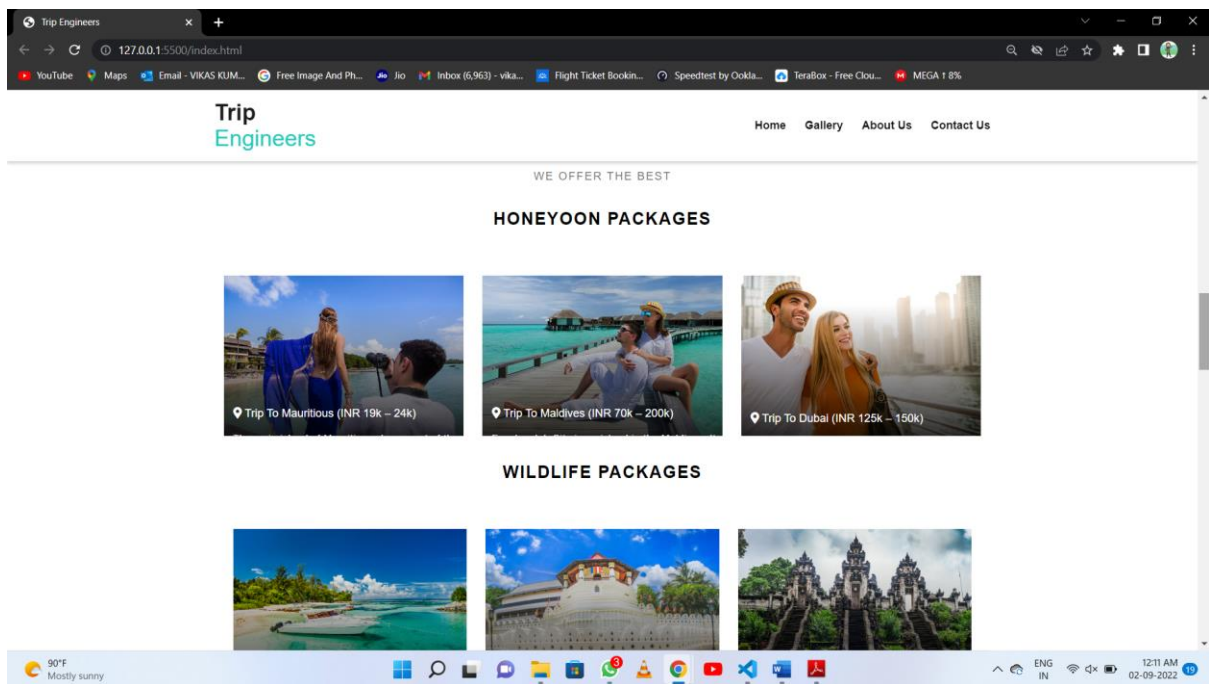
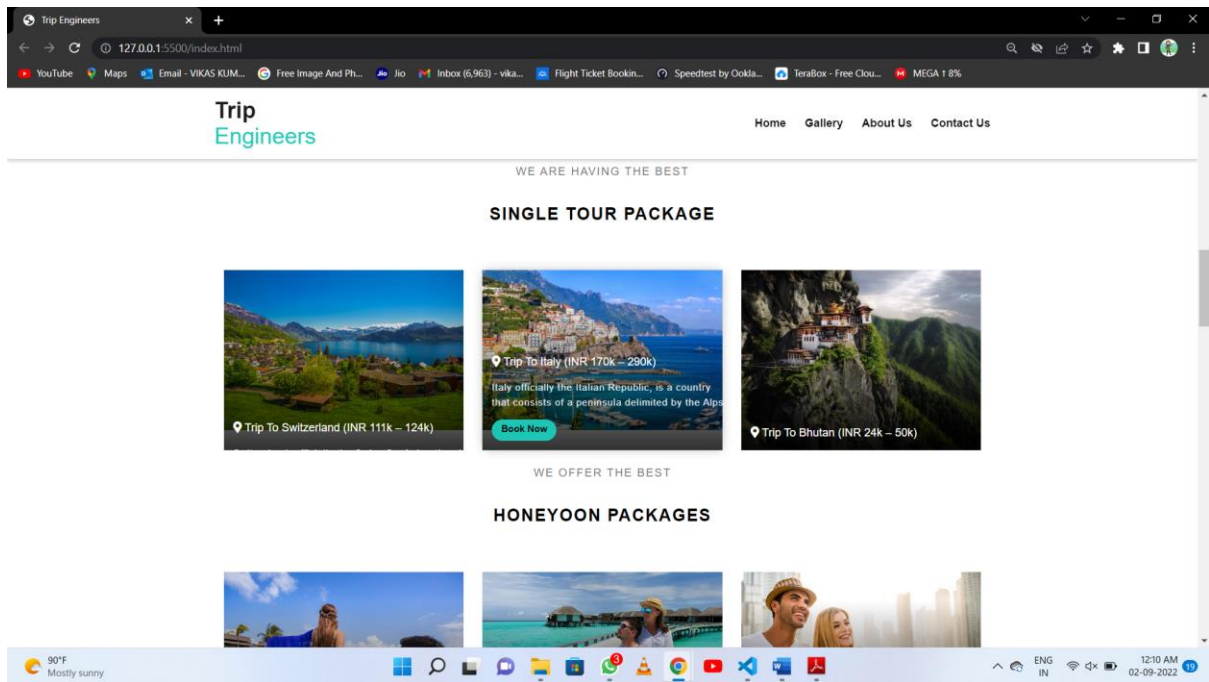
- Brief description and name of each module
- An outline about the functionality of every module
- Interface relationship and dependencies between modules
- Database tables identified along with their key elements
- Complete architecture diagrams along with technology details

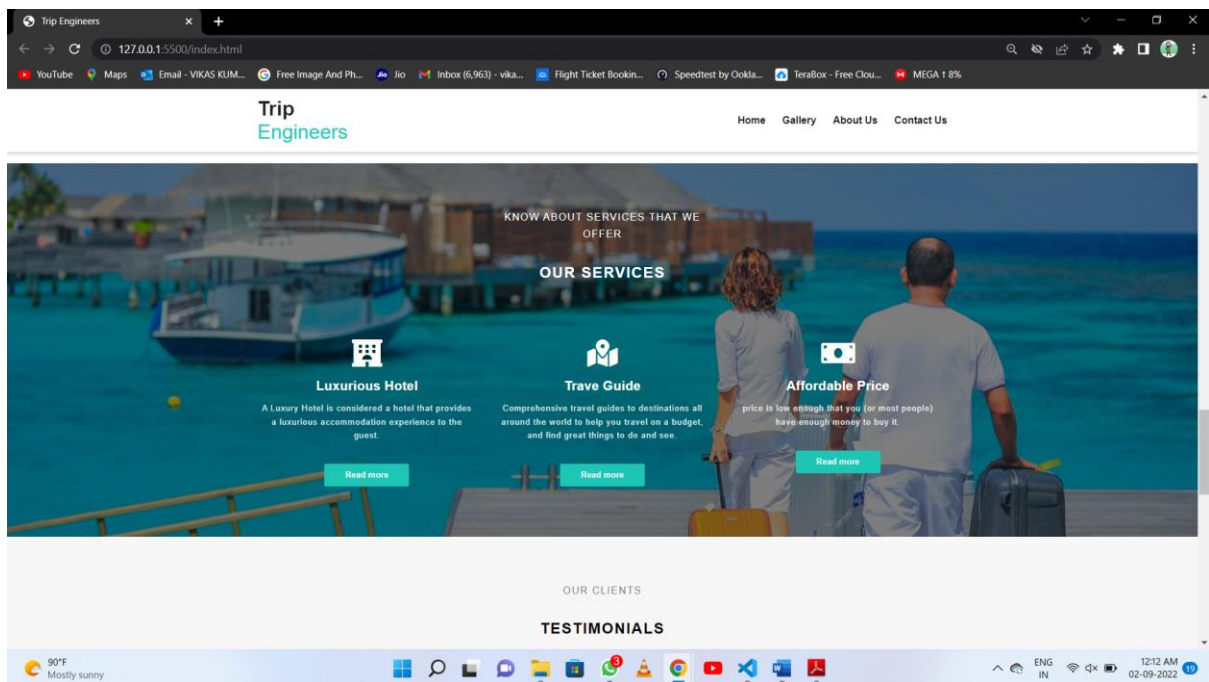
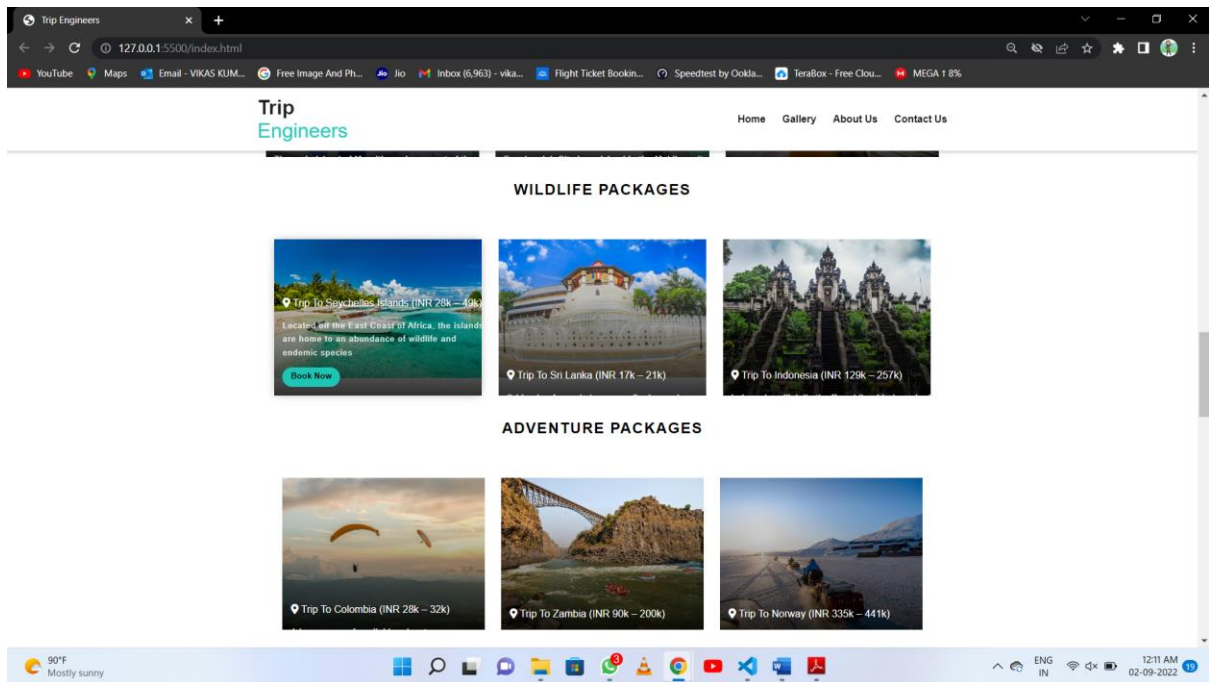
Low-Level Design (LLD)

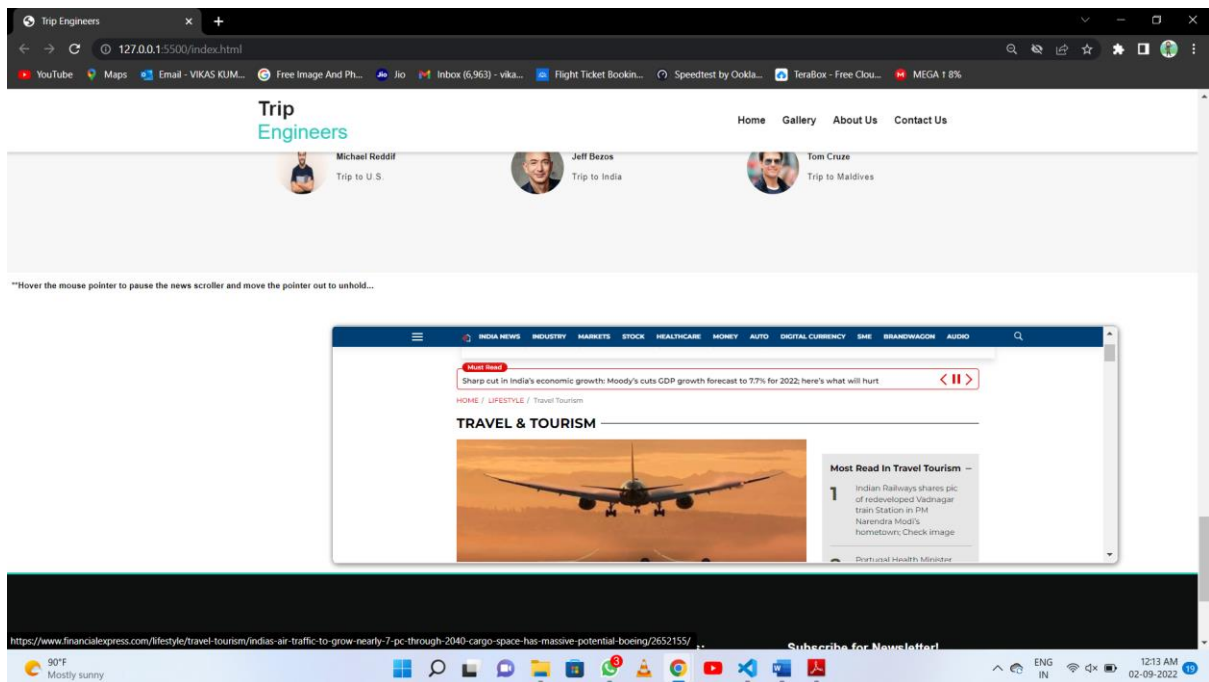
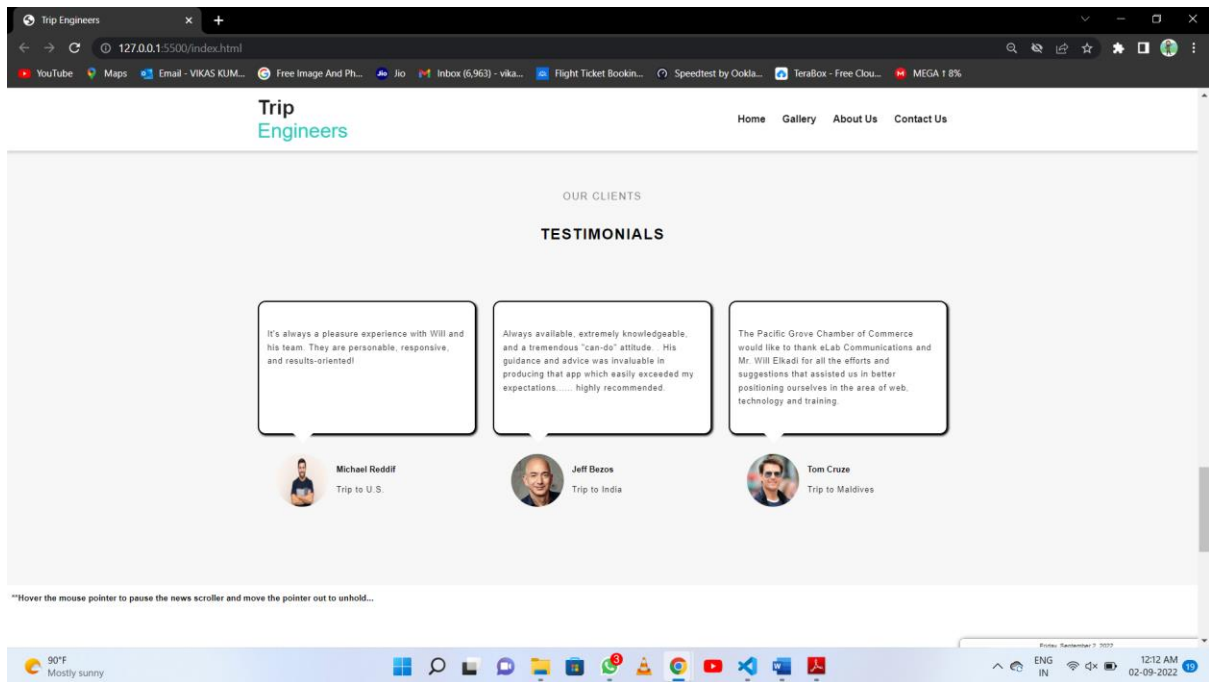
- Functional logic of the modules
- Database tables, which include type and size
- Complete detail of the interface
- Addresses all types of dependency issues
- Listing of error messages
- Complete input and outputs for every module

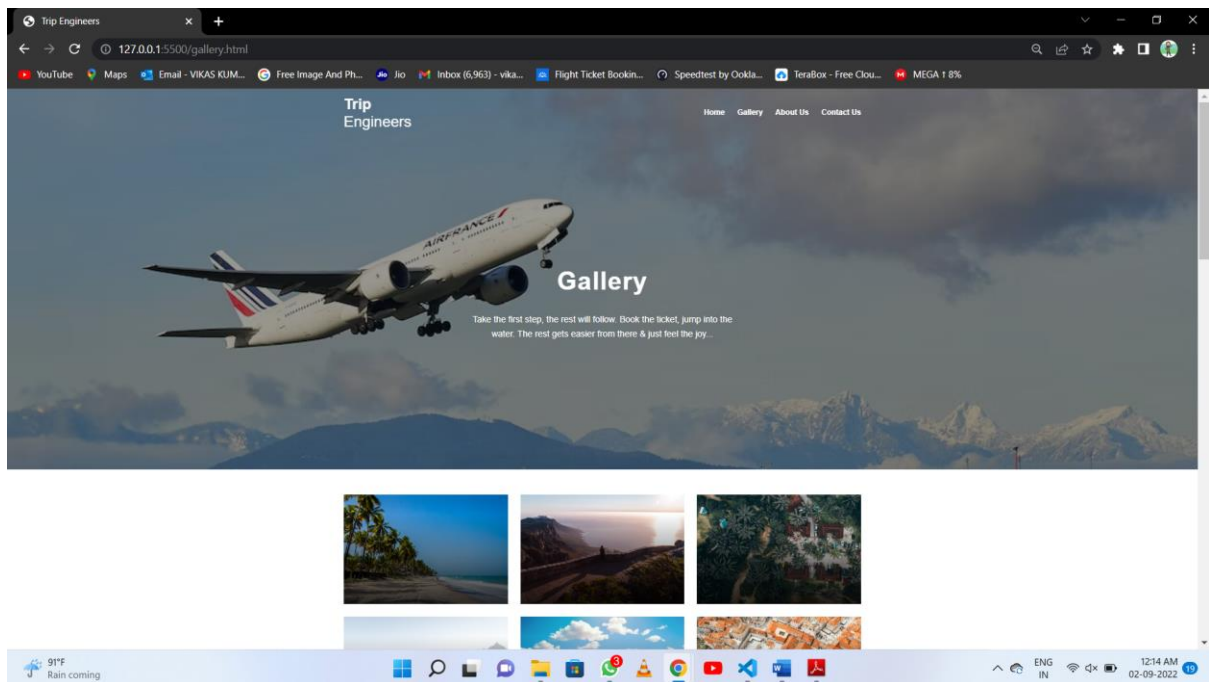
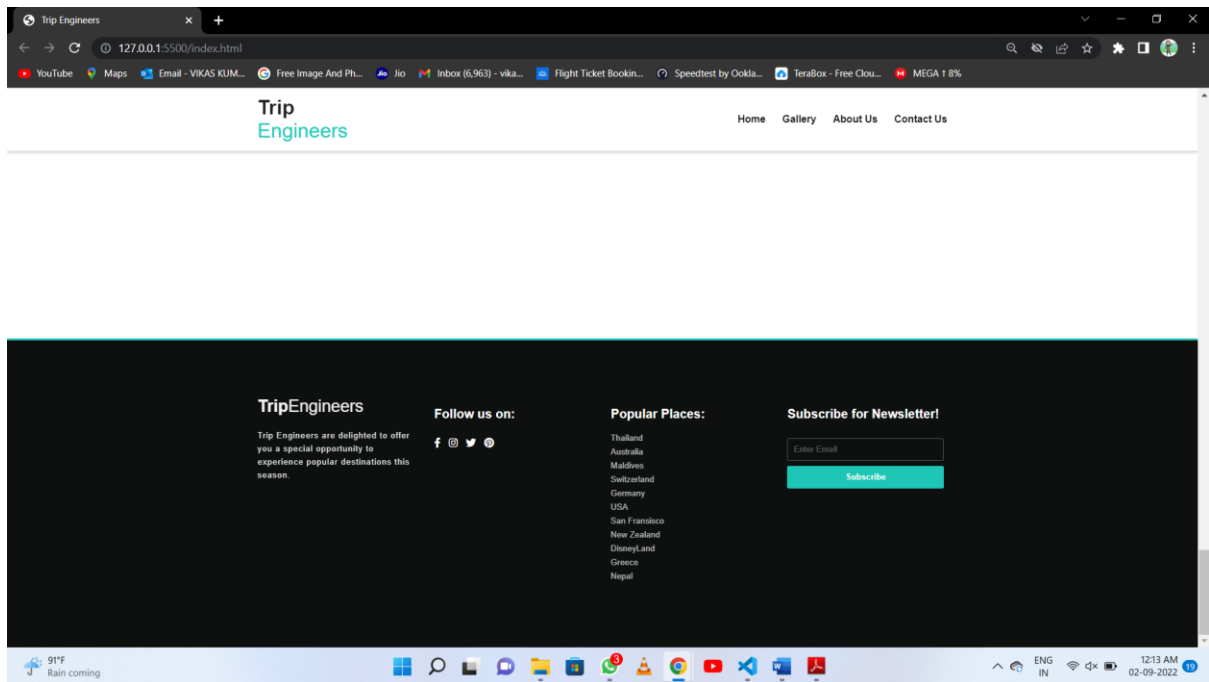
Screenshots

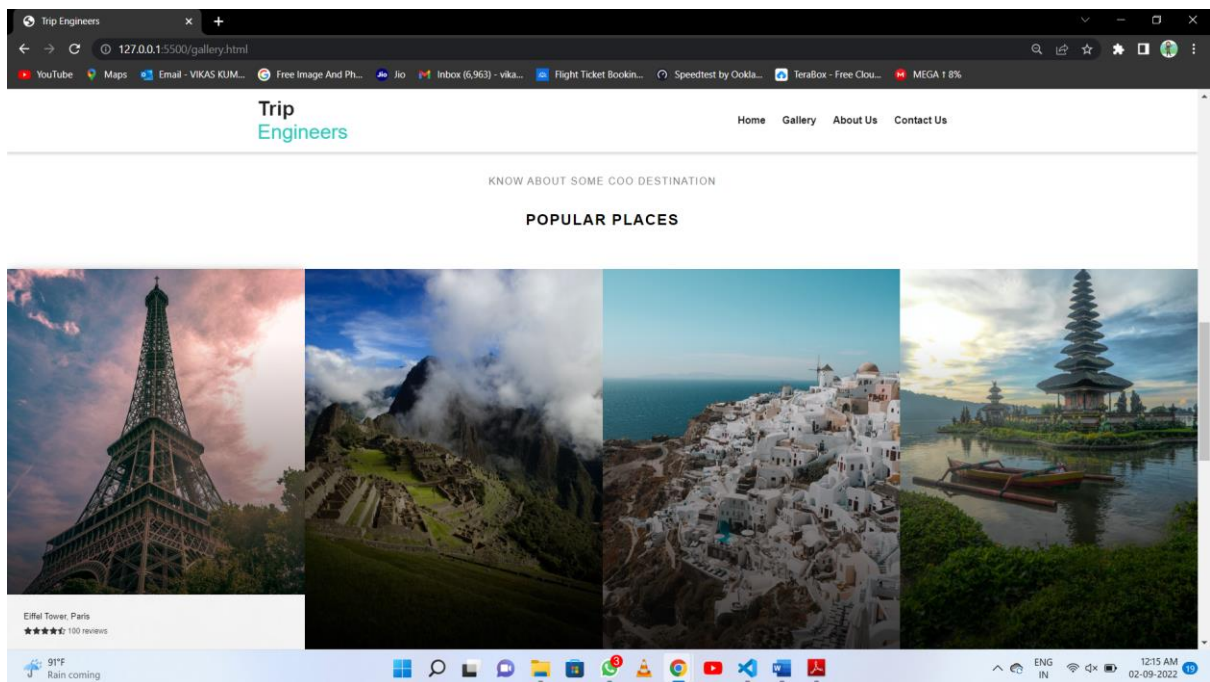
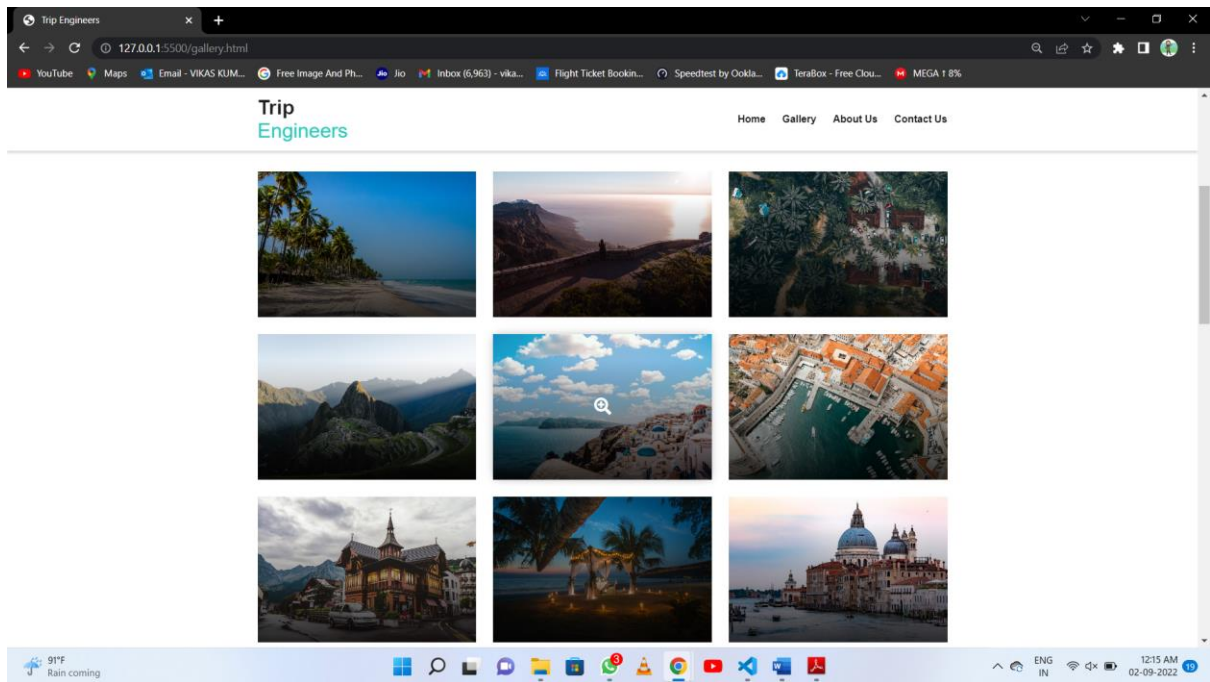


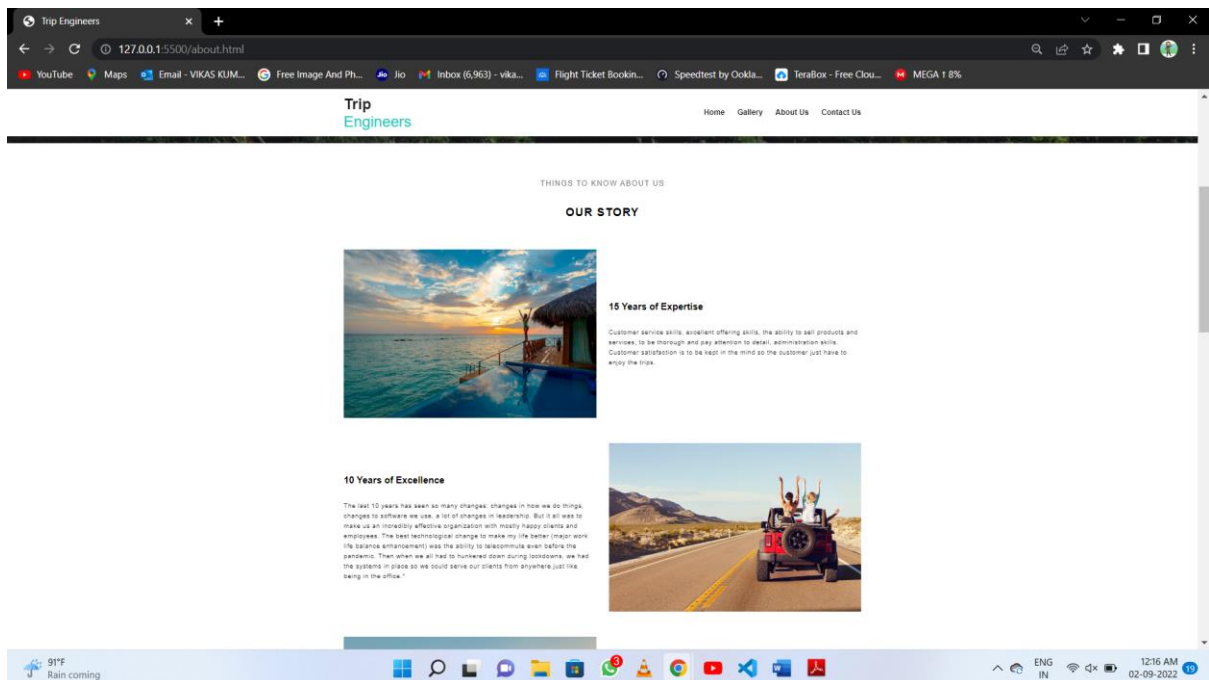
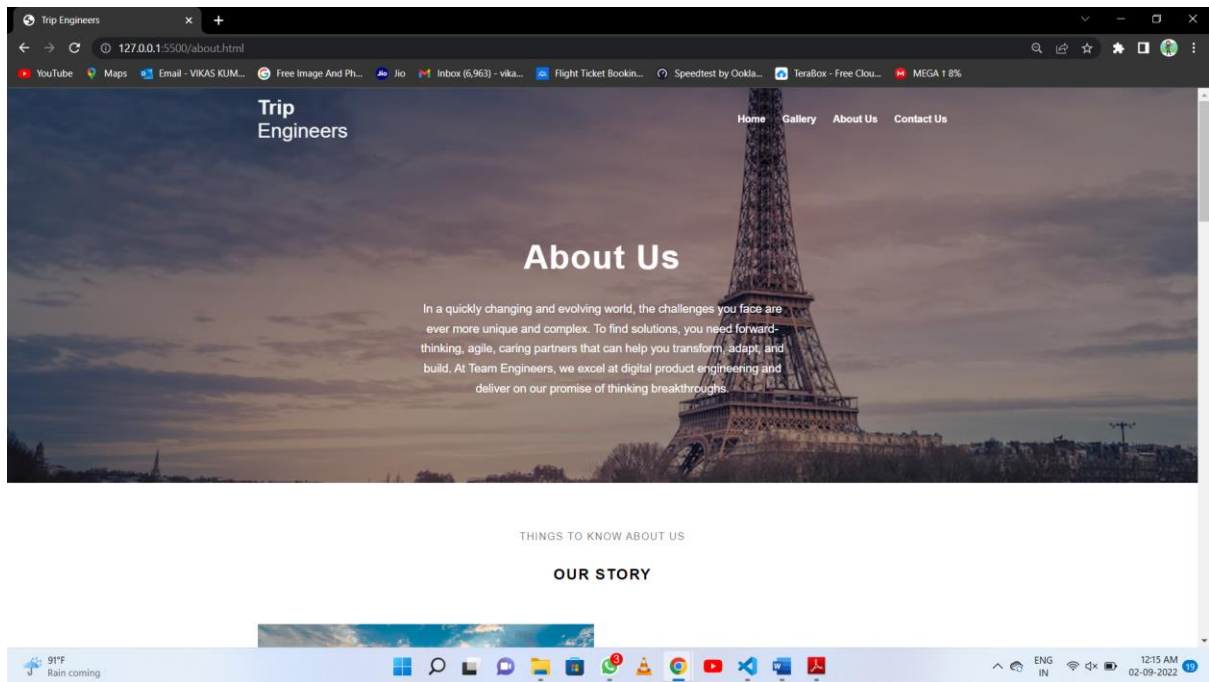


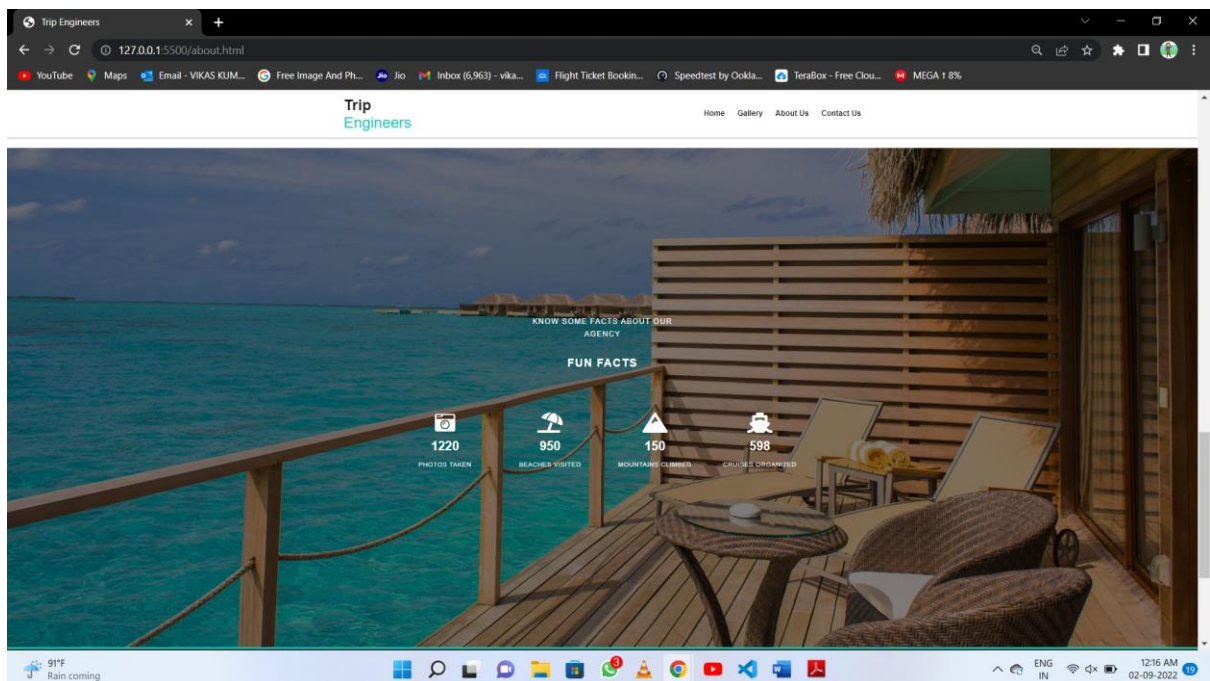
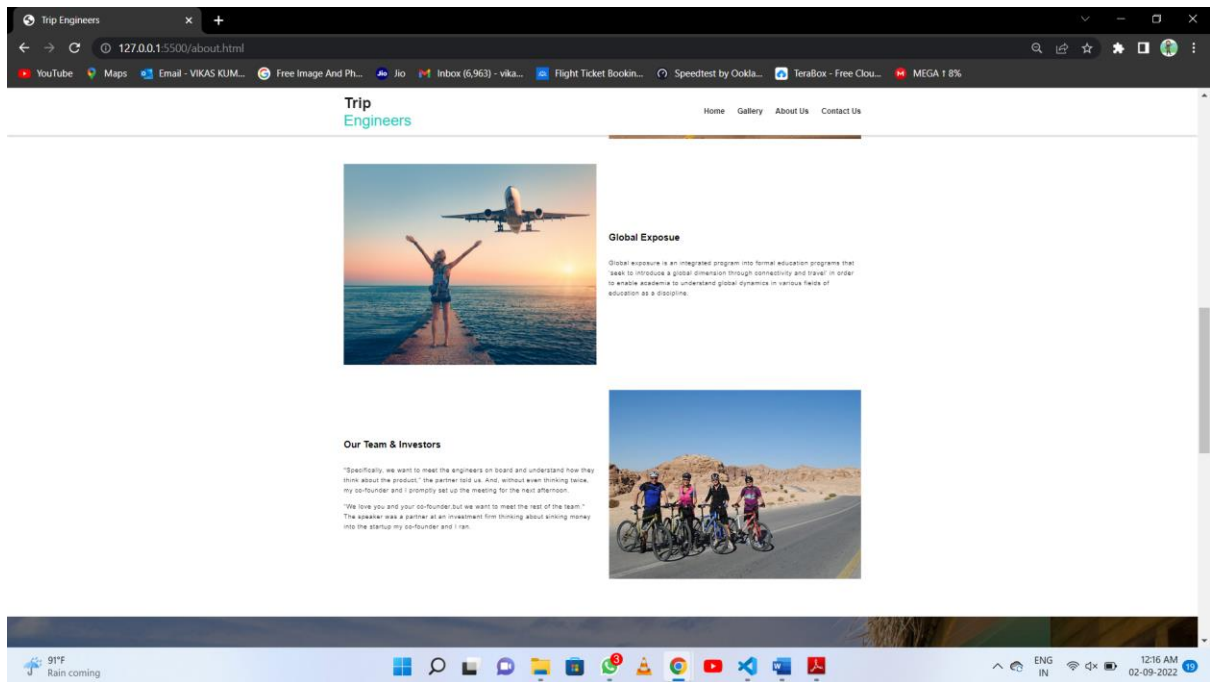


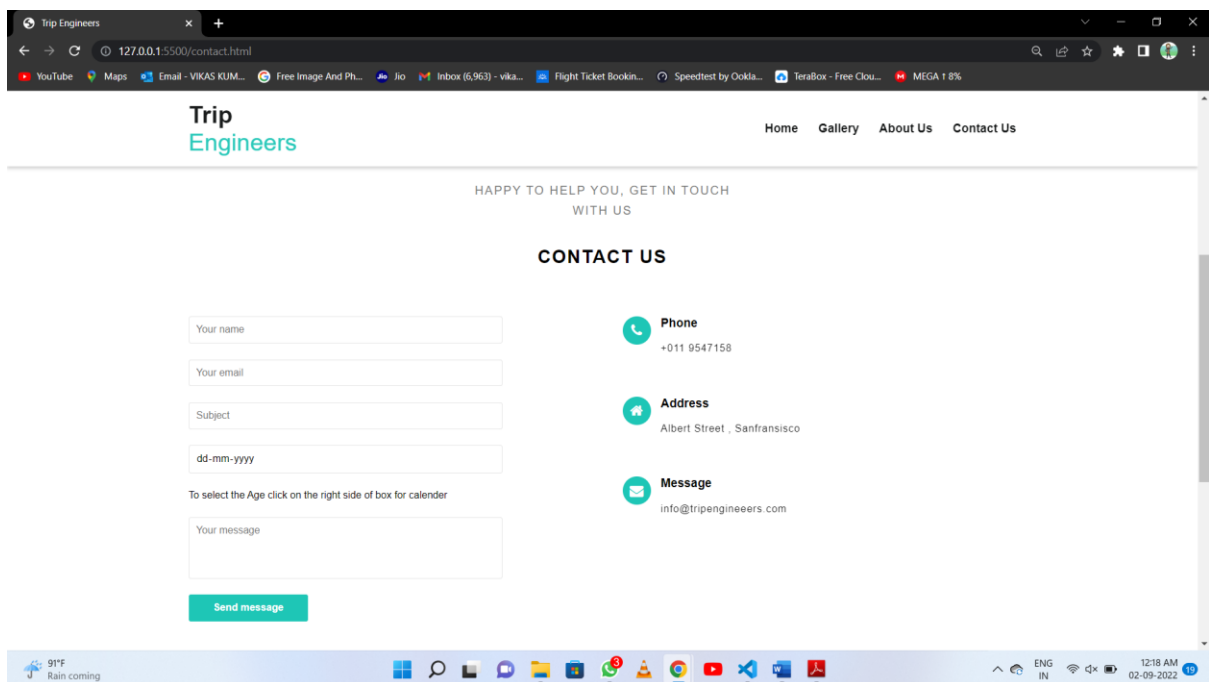
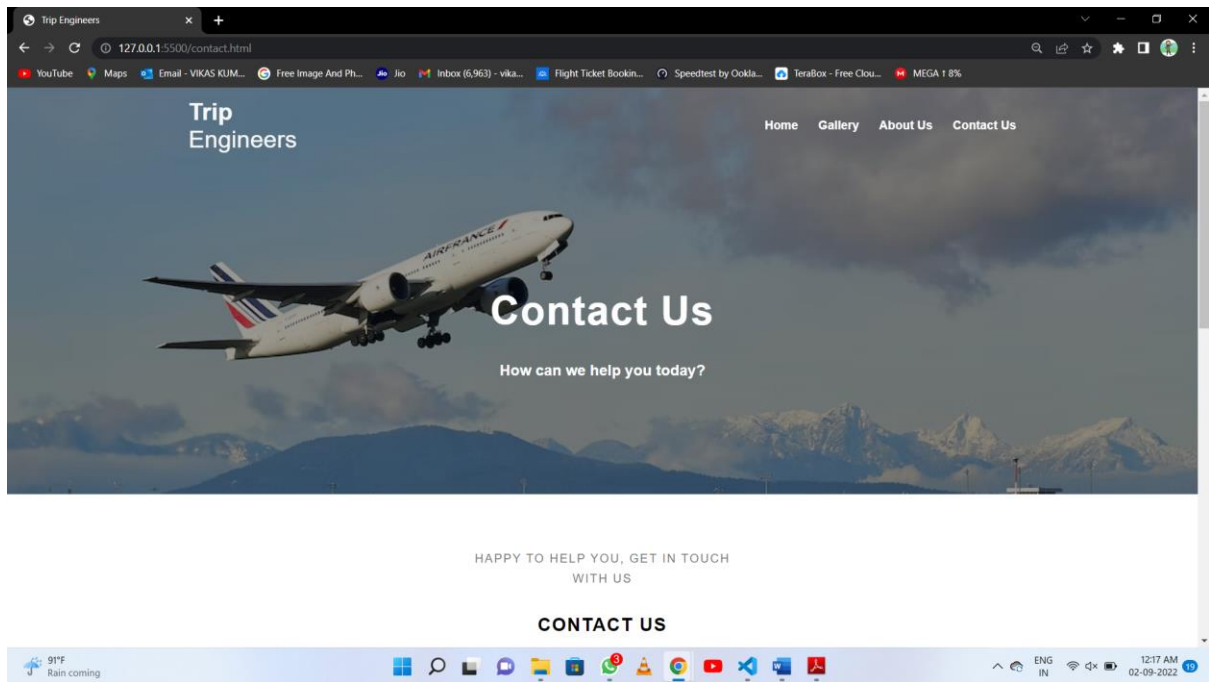












Phase 4: Testing

Testing Objective

In this section the functionalities will be verified and validated. This helps the developer to identify and understand the limitation and possible vulnerabilities of the web application.

Testing Design

White Box Testing

White Box Testing is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security. In white box testing, code is visible to testers so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing and Glass box testing.

The term “WhiteBox” was used because of the see-through box concept. The clear box or WhiteBox name symbolizes the ability to see through the software’s outer shell (or “box”) into its inner workings.

White box testing involves the testing of the software code for the following:

- Internal security holes
- Broken or poorly structured paths in the coding processes
- The flow of specific inputs through the code
- Expected output
- The functionality of conditional loops

- Testing of each statement, object, and function on an individual basis

The testing can be done at system, integration and unit levels of software development. One of the basic goals of whitebox testing is to verify a working flow for an application. It involves testing a series of predefined inputs against expected or desired outputs so that when a specific input does not result in the expected output, you have encountered a bug.

Black Box Testing

Black Box Testing is a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths. Black Box Testing mainly focuses on input and output of software applications and it is entirely based on software requirements and specifications. It is also known as Behavioral Testing.

- **Requirement** – This is the initial stage of SDLC and in this stage, a requirement is gathered. Software testers also take part in this stage.
- **Test Planning & Analysis** – Testing Types applicable to the project are determined. A Test Plan is created which determines possible project risks and their mitigation.
- **Design** – In this stage Test cases/scripts are created on the basis of software requirement documents
- **Test Execution**– In this stage Test Cases prepared are executed. Bugs if any are fixed and re-tested.

HAPPY TO HELP YOU, GET IN TOUCH
WITH US

CONTACT US

Your name

raju

Please include an '@' in the email address. 'raju' is missing an '@'.

dd-mm-yyyy

To select the Age click on the right side of box for calendar

Your message

Send message



Phone

+011 9547158



Address

Albert Street , Sanfransisco



Message

info@tripengineers.com

Phase 5: Maintenance

Once the system is deployed, and customers start using the developed system, following 3 activities occur

- Bug fixing – bugs are reported because of some scenarios which are not tested at all
- Upgrade – Upgrading the application to the newer versions of the Software
- Enhancement – Adding some new features into the existing software

The main focus of this SDLC phase is to ensure that needs continue to be met and that the system continues to perform as per the specification mentioned in the first phase.