**Gyan Ganga Institute of Technology and Sciences**

**Jabalpur, Madhya Pradesh**



**Major Project 2 Report**

**TITLE**

MONTH 2021

| Aaryan Soni | 0206CS181001 |
| --- | --- |
| Adarsh Singh Raghuwanshi | 0206CS181011 |
| Aman Jaisinghani | 0206CS181017 |
| Deepankar Dubey | 0206CS181047 |
| Dev Khare | 0206CS181050 |
| Dwij Jain | 0206CS181056 |

**Certificate**

This is to certify that Aaryan Soni, Adarsh Singh Raghuwanshi, Aman Jaisinghani, Deepankar Dubey, Dev Khare and Dwij Jain, students of class CS-1 have worked on their project entitled “TITLE” under my supervision. This project is approved for submission towards partial fulfilment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering.

| Dr. Ashish Mishra | Dr. Ashok Verma |
| --- | --- |
| Faculty, Computer Science and Engineering | Head of Department,  Computer Science and Engineering |

**Declaration**

We hereby declare that the project entitled “TITLE” submitted to Gyan Ganga Institute of Science and Technology, Jabalpur, is a record of original work done by our team under the guidance of Dr. Ashok Kumar Verma, Head of Department, Computer Science and Engineering.

This project is submitted in the partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering. The results embodied in this thesis have not been submitted to any other University of Institute for the award of any other degree of any degree of diploma.

**Acknowledgement**

We are grateful to Dr. Ashok Kumar Verma, Head of Department of Computer Science Engineering, Gyan Ganga Institute of Science and Technology, Jabalpur, for providing every help in accomplishing to its final stage the project on “TITLE” within stipulated time.

The components of this software have been developed in consultation with my associate faculty and we have been greatly benefited by their thoughtful and constructive suggestions, for which we are further thankful to them.

Aaryan Soni

Adarsh Singh Raghuwanshi

Aman Jaisinghani

Deepankar Dubey

Dev Khare

Dwij Jain

**Table of Content**

| 1 Introduction | | | 6 |
| --- | --- | --- | --- |
|  | 1.1 Purpose | | 6 |
|  | 1.2 Scope | | 6 |
|  | 1.3 Target Audience and Assumptions | | 7 |
| 2 System Overview | | | 7 |
|  | 2.1 Functional Requirements | | 7 |
|  | | 2.1.1 Hardware Requirements | 8 |
|  | | 2.1.2 Software Requirements | 8 |
|  | | 2.1.3 User Interface Requirements | 8 |
| 2.1.4 Interactive Components | 9 |
| 2.1.5 Non-Interactive Components | 9 |
|  | 2.2 Non-Functional Requirements | | 9 |
|  | | 2.2.1 Usability Requirements | 9 |
| 2.2.2 Security Requirements | 10 |
| 2.2.3 Performance Requirements | 10 |
| 2.2.4 Reliability Requirements | 10 |
| 2.2.5 Project Methodology | 11 |
| 3 Team Structure and Organisation | | | 12 |
|  | | 3.1 Team Members and Roles | 12 |
|  | | 3.2 Duration | 13 |
| 4 Diagrams | | | 14 |
| 5 Software Testing | | | 18 |
| 6 Business Diagram | | | 20 |
| 7 Future Prospects | | | 21 |
| 8 Conclusion | | | 21 |
| 9 References | | | 22 |

1. **Introduction**
   1. Purpose
   2. Scope

1.3 Target Audience and Assumption  
Anyone with some programming experience, with familiarity in Python and C, can understand this document.

The document is intended for developers, software architects, testers, project managers and documentation writers.

1. **System Overview**

This section is an overview of the main features of the project. Requirements are structured by functionality.

* 1. Functional Requirements
* UI: A user interface will help the user to interact with the application and make corresponding choices.
* Application: Provides a platform where the software and the hardware interact with each other in order to deliver services to the user.
  + 1. Hardware Requirements  
         
       PC with minimum Pentium Dual Core.  
       RAM more than 128 MB.  
       Keyboard and mouse or any equivalent device.
    2. Software Requirements
    3. User Interface Requirements

User interface requirements are briefly mentioned below:

1. Content presentation

2. Easy Navigation

3. Simple interface

4. Responsive

5. Consistent UI elements

6. Default settings

7. User centric approach

* + 1. Interactive Components
    2. Non-interactive Components

Non-interactive components of this project are icons, background colours, button shapes and sizes, and other static components required to make the application work smoothly.

* 1. Non-functional Requirements

This section describes the non-functional requirements of the project. The non-functional requirements are to secure against attackers, the application should be easy to use, automatic, universally available and for individuals.

* + 1. Usability Requirements

The application has a simple interface which is intuitive to use for the user. Though it does not have the latest design guidelines, it still provides all the basic functionality for user interaction. The system developed is intended to be used by the general public and should not have a steep learning curve. Also, anyone around the globe can access the application with no barriers.

* + 1. Security Requirements
    2. Performance Requirements

2.2.4 Reliability Requirements

Reliability is an attribute which specifies how likely the system or its element would run without a failure for a given period of time under predefined conditions.

2.2.5 Project Methodology

Agile and Scrum software development approach was used to develop this project. It advocates adaptive planning, evolutionary development, early delivery, and continual improvement, and it encourages flexible responses to change. Scrum is one of the methodologies that follow Agile principles. It is a combination of iterative and incremental models. Agile focuses on a single task with daily collaboration and communication and focuses on fast delivery.

Agile software development model, when compared to traditional software development models is fast, flexible and cost effective. By clarifying an acceptance criterion, estimation and breakdown of user stories, tasks can be broken down that can fit in a spirit cycle.

In the spirit pre planning phases of this project, user stories were estimated and broken down into Themes, Epics, Stories and Tasks. A task was assigned to each team member. The tasks when completed were reviewed, tested and then used as a base for the next spirit cycle.

Agile software development approach enabled us to have small scale focuses with rapid development cycles and minimised resources allocated to testing.

**3 Team Structure and Organisation**

3.1 Team Members and Roles

| **Name** | **Roles** | |
| --- | --- | --- |
| Aaryan Soni | Backend | Flask and SMTP |
| Adarsh Singh Raghuwanshi | AWS EC2 |
| Dwij Jain | Twillio API |
| Aman Jaisinghani | Frontend | HTML, CSS |
| Deepankar Dubey | Bootstrap |
| Dev Khare | Documentation and Testing | |

**Team 1**

The task of team 1 is to develop the backend functionality of the project.

**Team 2**

The assigned task of team 2 was to develop the frontend of the website

**Team 3**

**Documentation and Testing**: The responsibility of Dev Khare was to keep documenting the progress made by the team, handle version control, and continuously integrate progress made after testing code.

3.2 Duration

| **SNo.** | **Project Phase** | | **Working Days** |
| --- | --- | --- | --- |
| 1 | Business Requirements | | 7 |
| 2 | Application Design | | 7 |
| 3 | Development and Testing | | 15 |
| 4 | Documentation | | 7 |
| 5 | Deployment User Acceptance Testing | | 4 |
| 6 | Project Management | | 5 |
|  | | Total | 45 |
|  | | |  |

1. **Diagrams**

Use Case Diagram

| Sequence Diagram  Data Flow Diagram Level 0  Data Flow Diagram Level 1  Class Diagram |
| --- |

| Entity Relationship |
| --- |
|  |
|  |

**5 Software Testing**

Testing is the process of executing a program with the aim of finding errors. We perform testing of our system before deployment for efficient functioning of our system.

Types of Testing:

1. Functional Testing  
   It ensures that the application is functioning correctly. This type of testing focuses on the main purpose and flow of the application, ensuring that all its features are responsible and meet specifications
2. Usability Testing  
   Usability testing is a technique used in user-centred interaction design to evaluate a product by testing it on users. This can be seen as an irreplaceable usability practice, since it gives direct input on how real users use the system. It is more concerned with the design intuitiveness of the product and tested with users who have no prior exposure to it. Such testing is paramount to the success of an end product as a fully functioning application that creates confusion amongst its users will not last for long.
3. Compatibility Testing  
   Compatibility testing is a part of non-functional testing conducted on application software to ensure the application's compatibility with different computing environments.

**Bot Testing**

| **Module** | **Scenario** | **Expected Result** | **Result** |
| --- | --- | --- | --- |
| User | User initiates chat with bot | Bot connects | Success |
| User sends message | User provides complete details | Message is received | Success |
| Return message: success | Mail is successfully forwarded | Mail sent | Success |
| Return message: fail | Mail could not be forwarded | Mail not sent (internet connectivity was severed) | Success |

**Web Interface Testing**

| **Module** | **Scenario** | **Expected Result** | **Result** |
| --- | --- | --- | --- |
| User | Visits URL | Website loads | Success |
| Enters details | Details are recorded | Data entered | Success |
| Return message: success | Mail is composed | Mail is sent | Success |
| Return message: fail | Mail is not forwarded | Mail is not sent | Success |

**6 Business Model**

Business Process Modelling Notation (BPMN) is a flow chart method that models the steps of a planned business process from end to end. A key to Business Process Management, it visually depicts a detailed sequence of business activities and information flows needed to complete a process.

Its purpose is to model ways to improve efficiency, account for new circumstances or gain competitive advantage. The method has been undergoing a standardisation push in the past few years and is now often called by a slightly different name: **Business Process Model and Notation**, still using the BPMN acronym. It differs from Unified Modelling Language (UML) used in software design.

BUSINESS DIAGRAM

1. **Future Prospects**

**8 Conclusion**

**9 References**

1 Twilio Documentation

<https://www.twilio.com/docs/api>

2 Flask Documentation

<https://flask.palletsprojects.com/en/2.0.x/tutorial/index.html>

3 Ngrok GitHub Repository

<https://github.com/inconshreveable/ngrok>

4 AWS Documentation

<https://docs.aws.amazon.com/general/latest/gr/Welcome.html>

<https://docs.aws.amazon.com/ec2/index.html>