Readme :-

- **How to start Contact Manager :-**

    1. **Use this link to open the web application :- https://seprojcontactmanager.herokuapp.com**

    2. **Also you can run project on local host using zipped codes :-**
        a. **Unzip the codes.**
        b. **Place it in a folder and open the folder with VS code editor or any other editor which creates a python(3.8.8) environment.**
        c. **Run on terminal : pip install -r requirements.txt**
        d. **Run on terminal : python manage.py makemigrations**
        e. **Run on terminal : python manage.py migrate**
        f. **Run on terminal : python manage.py runserver**

Now after running the previous command, you are given link to local host the software project.

- **How to use Contact Manager :-**
  You can also register from the app and we are also attaching Login credentials :
  Username : Test_Login
  Password : just_for_testing

    - You have to have a login in credential in order to login into the website.
    - After registering/logging into the web application you now proceed to manage contacts.
    - You have all the functionalities to manage your contacts like add contacts, delete contacts,update contacts,linking with

  email and phone number, search functionality by name,info,etc.
- It is very useful to preserve contacts with your login credentials.

- **Contribution by each member**
  - **Contribution in Code**

Team Member	**Work done**
Mohit Ahirwar	View.py- Home Page view,Contact detail view,Contact create,update and delete view Models.py Templates and static files Html files-base.html,create.html,delete.html manage.py
Ram Khandelwal	Views.py-Search view, Authentication view,Contact Admin module Urls.py Requirements.txt Settings.py Html files-index.html,search.html,update.html settings.py

  - **Contribution in Testing-**

Team Member	**Work done**
Mohit Ahirwar	Automation Scripts running on Selenium and testing on login and adding contacts.
Ram Khandelwal	Unit, Integration and acceptance testing on the software project

  - **Contribution in documentation**

Team Member	**Work done**
Mohit Ahirwar	SRS-Functional Requirements,Performance Requirements,Design Constraints,Schedule and budget estimates,Work flow,Use case,User scenario

	Design & Architecture-Use case Scenario,Physical view,Process view,Admin Page and Authentication wireframe Essence Report-Contents,Sprints
Ram Khandelwal	SRS-Context,Problem,Scope,Interface Requirements,Non-functional requirements,Measures,description Design & Architecture-Quality Attribute focussed,Logical view,Development view,Home page and search wireframe Essence Report-Use case,Microservices,Kernel alphas states

  - **Other processes such as requirement gathering,deployment,meetings and discussions,starting with django,readme file etc, have equal contribution by both the members.**

- **Summary**

So finally our project finishes with all the proper documentation,codes and a successfully deployed application.Any changes in the application further in the future are most welcome.It was a amazing experience for both of us to be a part of software engineering project which follows software engineering principles.

We would like to express our gratitude by giving special thanks to Dr. Sumit Kalra sir who taught us how to use software engineering principles in software development.