USC TeleHELP

Official Documentation

Table of Contents

Logistics

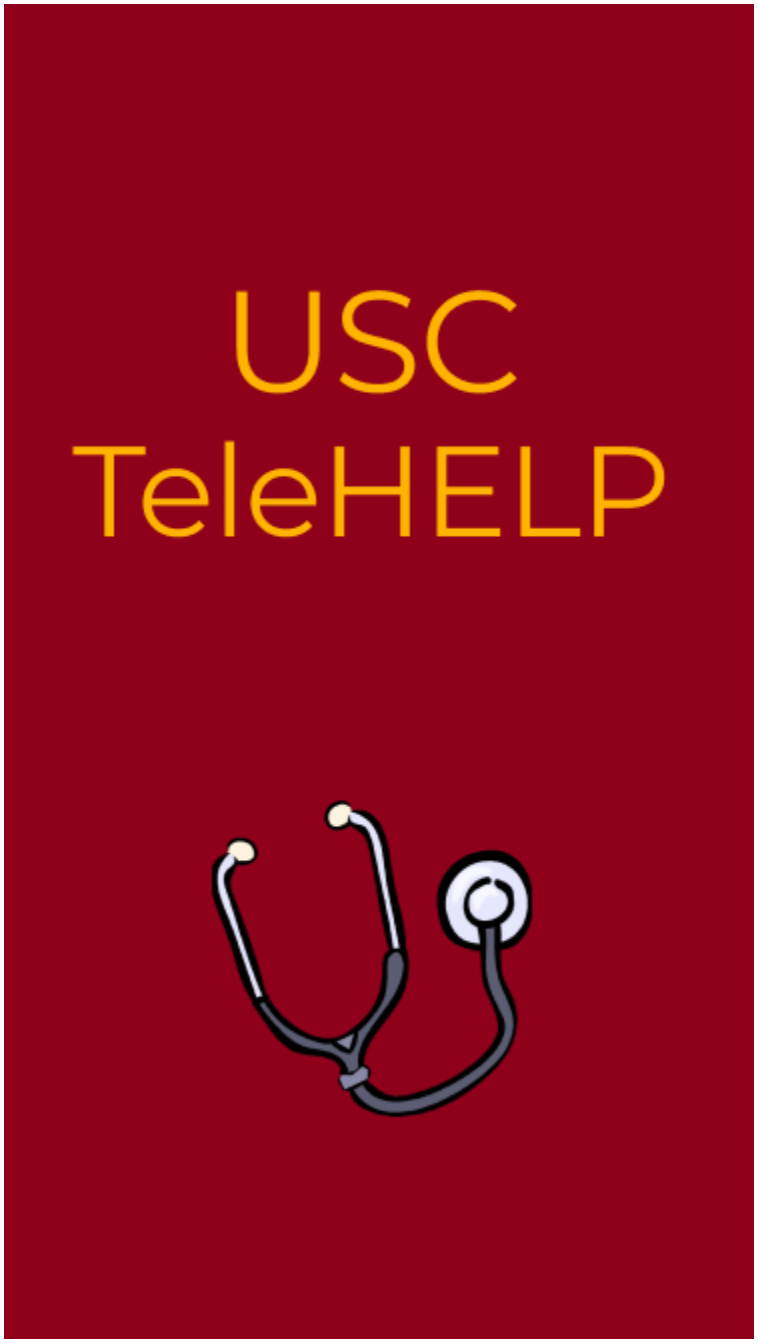
Project Source Material

- Github: https://github.com/atchesonUSC/csci401_project

CSCI-401 Team Contact Information

- Samuel Atcheson (email: atcheson@usc.edu)
- Emily Liu (email: liu246@usc.edu)
- Peng Gan (email: penggan@usc.edu)
- Sarah Okamoto (email: saokamot@usc.edu)

User Documentation



Background

The purpose of this documentation is to overview the development, setup, and usage specifications of the USC TeleHELP application. USC TeleHELP is a novel technology that enables users with telemedicine appointments to visit a one-stop-shop on how to be prepared for telemedicine at Keck. Catering to audiences with a wide range of technological proficiencies, USC TeleHELP is intended to make telemedicine through its companion, USC TeleCARE, more accessible. This project has been developed as a collaboration between stakeholders at USC Keck Medical Center and a student development team representing the Spring 2021, CSCI-401 Capstone course. For further information regarding the usage and deployment of USC TeleHELP, please refer to the below sections.

Problem Statement

Over the last several years, telemedicine has grown in popularity for its capacity to allow individuals the receipt of expert medical care when the option of appearing at an in-person appointment may be infeasible. As such, video conferencing technology has provided an avenue for connecting patients with physicians. At the Keck Medical Center, a proprietary application -- USC TeleCARE -- has been employed to enable telemedicine visits.

For senior patients or individuals who may not be technologically proficient, the task of accessing one's virtual appointment can present a daunting challenge. Some of these complexities might include: installing the USC TeleCARE app, making sure one's camera and microphone are enabled for use during the appointment, etc. These challenges, as well as others, may create a barrier-to-access for less technologically proficient users, potentially obstructing individuals from receiving medical attention.

Solution and Goals

The purpose of this project is to develop a supportive technology for allowing users with varying degrees of technological proficiency to easily access telemedicine appointments through the USC TeleCARE application. In doing so, our team has developed a web-based application that offers users the ability to proactively seek real-time guidance on various aspects of the telemedicine appointment process. In this way, USC TeleHELP operates on a helpdesk-like service model. USC TeleHELP provides an easy-to-use interface that supports simple interactions - we want users to have as streamlined and straightforward an experience as possible. In addition, because USC TeleHELP is a web-based application, users can access USC TeleHELP from any device on which they have a supported web browser, and are not restricted based on Android or iOS platforms.

User Interface and Experience

Part of the obstacles between patients and telemedicine is their unfamiliarity with hardware peripherals, touchscreens, and reading websites and app screens. Therefore, the user interface is designed to be easily understandable while the user experience focuses on clearly telegraphed feedback so that the user understands the cause and effect of their actions within the application.

GrandPad Interface

One approach the TeleCARE application takes from is GrandPad.

Figure #1: The GrandPad interface displays large icons.

Large font with large icons that users can intuitively know what they are looking for even with elders.

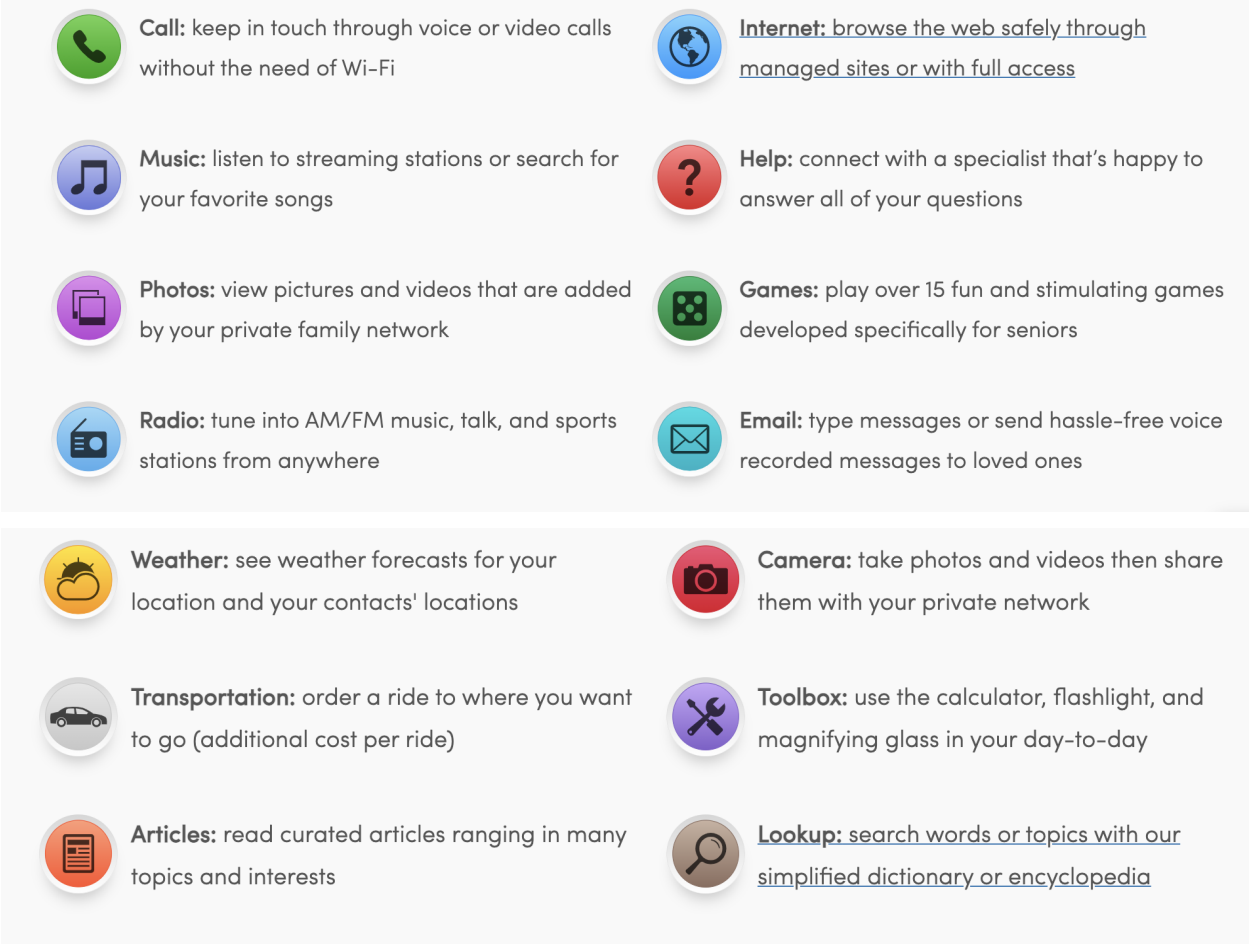


Figure #2: The GrandPad modules

Each icon identifies a specific interface and they can be known intuitively.
One issue is that they are on a specific tablet with a designed resolution. We designed a web app that's also mobile- and tablet-friendly so that most clients will already have a compatible device.

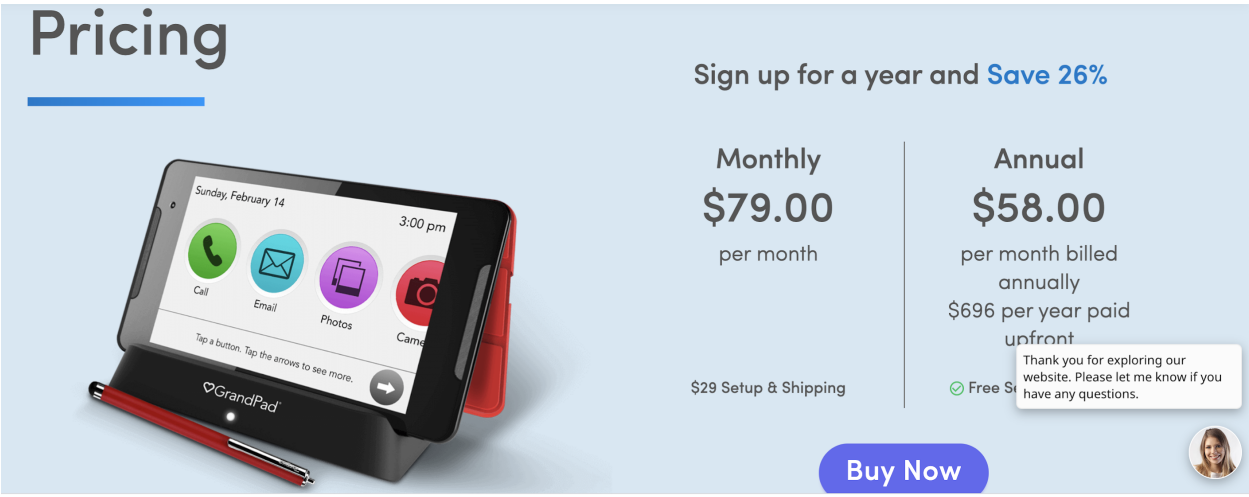


Figure #3: The pricing of GrandPad

They charge for a price and our app is complimentary for our users to download.

Modules

The TeleHELP application is made up of multiple “modules”, or guides covering the technical setup required for Keck telemedicine. The intent is to make telemedicine easier to approach by providing information and interactive steps. The modules are listed below.

Welcome

This module will provide the user with a brief overview of the USC TeleHELP and explain the relationship between TeleCARE and TeleHELP; namely, the latter’s role as assistive guide for preparing one’s technology and confidence and attending a virtual appointment through USC TeleCARE.

Installing USC TeleCARE

This module will provide users with information on how they can install the USC TeleCARE app on their device if they plan to use a tablet or smartphone for their appointment.

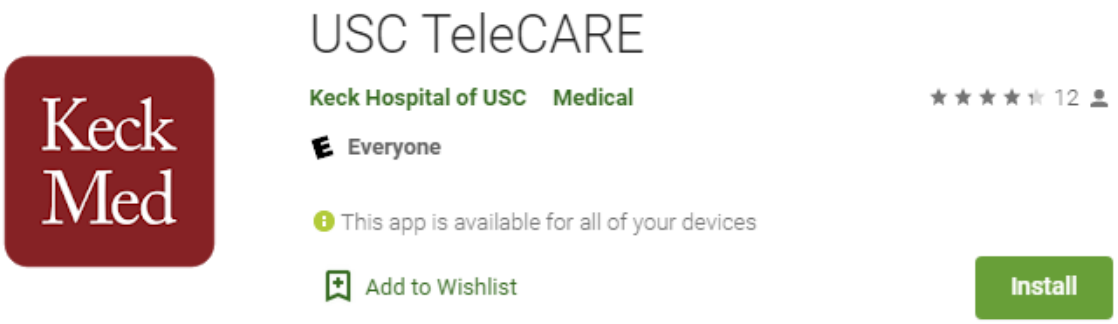


Figure #4: What the TeleCARE Play Store shows if the user’s Google Account has never installed the app on a device before.

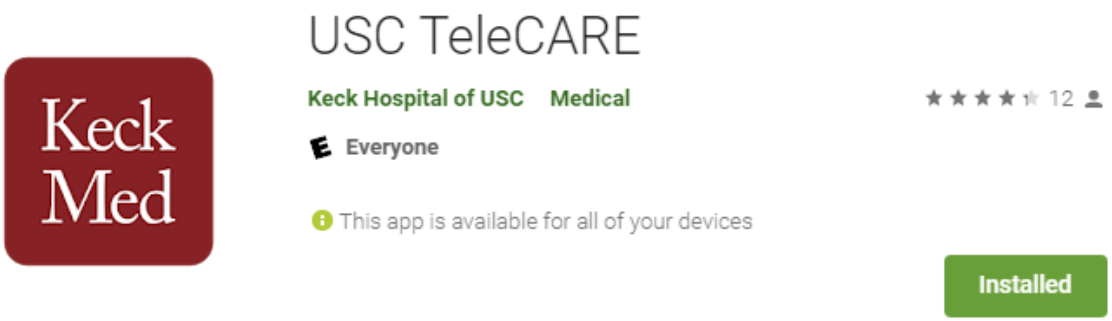


Figure #5: What the TeleCARE Play Store shows if the user's Google Account has at some point installed it on a device.

Browser Check

This module will help direct users to access their appointments through the USC TeleCARE-supported browsers (Google Chrome, Firefox, Edge, and Safari) if they plan to use a computer for their appointment visit. If they plan to use a smartphone or tablet, then they are guided to the *Installing USC TeleCARE* lesson.

Camera and Microphone Check

This module gives the user guidelines for using their camera. The aim is to ensure the user's camera is properly set up before they have an appointment. Step-by-step visual instructions tell the user how to enable their camera for the USC TeleCARE app for separate device operating systems (of which the options are: Android, and iOS). The module will also guide the user to different instructions for camera and microphone setup based on whether or not the user has used USC TeleCARE before. The user is also given tips on how to frame themselves so the doctor can properly see them.

This module also gives the user guidelines for using their microphone. The aim is to ensure the user's microphone is properly set up before they have an appointment. Step-by-step visual instructions tell the user how to enable their microphone for the USC TeleCARE app for separate device operating systems (of which the options are: Android, and iOS).

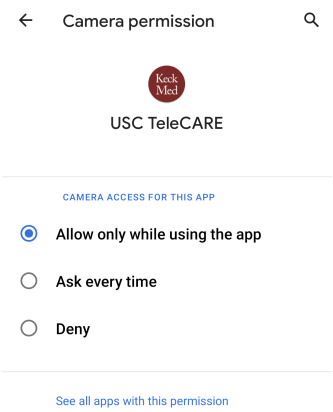


Figure #6: In Android, camera permissions for TeleCARE turned to “on”.

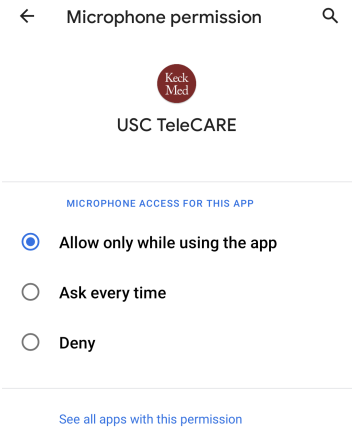


Figure #7: In Android, microphone permissions for TeleCARE turned to “on”.

Readiness Check

This module will advise users on how they can inquire with the USC TeleCARE Support Team on the setup of their technology (i.e. internet connection, camera, microphone, etc.).

Finding Your Waiting Room

This module will guide users through the steps of gaining access to USC TeleCARE through the “Enter Waiting Room” button provided in their appointment invitation email.

Using USC TeleCARE

This module will direct users through the steps of entering USC TeleCARE (covers access on computer platforms and smartphones/ tablets) and through the different stages of the waiting room access process.

Final Remarks

The root of TeleHELP’s purpose is to make telemedicine and the required technologies easier to get accustomed to through easy instructions and a simple interface. That said, the application itself is not a solution for every problem medical providers must tackle. Using TeleHELP itself requires some rudimentary handling of technology that patients may not have. Senior patients are a core demographic with a lack of technology proficiency that TeleHELP targets. While TeleHELP can be improved on to solve the knowledge and accessibility challenges patients have, lack of infrastructure (such as needing a router or acquiring a webcam) is something that requires in-person, third-party support and can be grown together or separately from a tool like TeleHELP.

Developer Documentation

Project Description

Telemedicine has become a vital component of health care delivery within the COVID-19 pandemic. Providers rapidly turned to telephone and video appointments in order to maximize physical distancing and protect those most vulnerable. As one of the most medically vulnerable patient groups, seniors have in particular stood to benefit from telemedicine. However, as many older individuals do not have video-enabled devices and/or the tech literacy to use them, the potential benefit of telemedicine for this patient population is instead emerging as a health disparity. This goal of this project is to create a senior-friendly web application (USC TeleHELP) that will aid this user group in navigating USC TeleCARE, USC's telemedicine service. The web application will consist of five modules to assist in any deficit in technological literacy for the USC TeleCARE service. The first module, "About USC TeleHELP" , details the purpose of the web application. The second module, "Camera and Microphone Help", provides users with information on how to enable both the camera and microphone on their device. The third module, "Internet Help", provides a phone number the user can call for inquiries about their device's internet connectivity. The fourth module, "How to Get USC TeleCARE", provides step-by-step instructions on how to access the USC TeleCARE service that the user will need for their upcoming telemedicine appointment. The fifth module, "Help Center", provides a phone number for any other issues the user may face when preparing for their telemedicine appointment. This application will be made compatible with Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari.

GitHub

The source material for this project is hosted in a public repository on GitHub. The following steps will guide you through how you can clone the repository to your local machine:

1. Visit the repository page on GitHub: https://github.com/atchesonUSC/telehelp
2. Navigate to the green "Code" button. Click on the button.
3. Click on the "HTTPS" option and copy the repository cloning URL.
4. Within a shell environment (Terminal on macOS and Command Prompt on Windows machines), run the following command from within the folder where you would like the repository to be located.

   $ git clone <repo-cloning-url>

   Where <repo-cloning-url> should be replaced with the URL copied from step 3.

5. The project repository has officially been cloned to your local machine.

Python

The source code for this project was written using primarily the Python scripting language. You will need to install the most recent version of Python 3 on your development machine in order to run the application.

If you are unsure of whether or not you have a version of Python 3 installed on your machine, or if you need to install Python 3, please refer to the following guide that offers a quick and easy walkthrough for version check and installation instructions.

Python 3 Setup Guide: https://realpython.com/installing-python/

PyCharm

For fast and easy development, PyCharm is a recommended environment for use with Python. PyCharm is a cross-platform IDE that is supported on macOS, Windows, and Linux platforms. The PyCharm Community edition is recommended, as it is free to use. In order to install PyCharm, please visit the official vendor site for installation instructions.

PyCharm Installation: https://www.jetbrains.com/help/pycharm/installation-guide.html

Pip

To install Python packages for enabling important development features, your machine must have the Python package manager installed, known as pip. The pip tool normally comes pre-installed on Windows and macOS platforms. You can check your current version of the pip by running the following command based on your operating system:

(Windows) *> pip -V*
(macOS) *$ pip3 --version*

Django

Django is a popular, powerful backend web framework in Python that enables rapid construction of engaging web applications. Django was chosen as the implementation framework of choice due to its flexibility, security, and scalable design features. More information regarding the use of Django can be found at the framework's official website (https://www.djangoproject.com/start/overview/).

An official release of the Django framework can be installed using the following command:

(Windows) \> py -m pip install Django
(macOS) $ python -m pip install Django

To verify that the installation of Django was successful, the following commands can be entered, and the return of a version number indicates a successful installation that can be seen by your machine's Python interpreter. After typing *python* into your system's shell environment, import the django module and attempt to print the version.

$ python
>>> import django
>>> print(django.get_version())
3.1

Bootstrap

Bootstrap is a widely-used frontend design framework, and was used for designing the USC TeleCARE user interface. In order to employ Bootstrap in your project development setup, you can elect to follow one of two approaches: (1) direct download or (2) BootstrapCDN (Content Delivery Network). Using BootstrapCDN is the recommended option, as it is more seamless and involves less file management.

In order to use Bootstrap in a templated HTML page, you should include the necessary <meta>, <link>, and <script> tags in the appropriate relative locations of your HTML file. To do so, please refer to the following webpage under the section entitled "Starter template" for guidance on the contents of the <meta>, <link>, and <script> tags necessary for enabling Bootstrap and where to include them.

This app uses Bootstrap 4.3.1.

Enabling Bootstrap content:
https://getbootstrap.com/docs/4.3/getting-started/introduction/

API

All lesson pages in the project (all apps minus the welcome app) extend one of these templates.

Lesson page template with 3 buttons
module_base.html

module_base.css

Lesson page template with 2 buttons
module_base_2b.html
module_base.css

Project structure

about_telehelp_app

Lesson 1: Welcome!
➔ about_telehelp_#.html

browser_check_app

Lesson 4: Browser check
➔ browser_check_#.html

camera_mic_app

Lesson 5: Camera and Microphone Check
➔ Cam_Mic_#.html

install_app

Lesson 2: Installing USC TeleCARE
➔ install_#.html

internet_check_app

Lesson 3: Internet Check
➔ internet_check_#.html

using_telecare_app

Lesson 4: Browser check
➔ using_telecare_#.html

waiting_room_app

Lesson 4: Browser check
➔ waiting_room_#.html

welcome_app

Welcome/home page

- ➔ welcome.html
- ➔ welcome.css

Launching USC TeleHELP for Development

In order to perform maintenance and development on the USC TeleHELP application, the Django framework offers support for a development server that will allow the site to be served from the localhost at port 8000. During this development stage, it is recommended to make make the following changings in the “telehelp/admin/settings.py” file:

```
DEBUG = TRUE
ALLOWED_HOSTS = []
```

On your machine’s command line, you may navigate to the “telehelp” directory, where there should be a script entitled “manage.py”. To launch the Django development server, you may execute the following command:

```
$ python manage.py runserver
```

From here, you may open a web browser and visit the following address:

http://127.0.0.1:8000/

This address will direct you to the app-under-development.

Launching USC TeleHELP for Production

To prepare the USC TeleHELP app for production, there are several settings that should be changed in the “telehelp/admin/settings.py” file. Within the base directory of the project, please execute the following command for checking the application’s settings before deploying into production.

```
$ python manage.py check --deploy
```

This command will show you the settings that you may want to consider changing to fit your desired deployment configuration.

Several options exist for hosting the USC TeleHELP application depending on preference, including: Amazon Web Services, Heroku, Python Anywhere, and Microsoft Azure. Hosting the application on an internally owned and operated server is also an option.

An additional setting that may be considered if user login functionality is to be implemented in the future would be the secret key setting, which is Django’s way of enforcing particular security features provided by the Django framework. If any form of user information is to be collected in a future implementation of the site, then the value of SECRET_KEY in the

“telecare/admin/settings.py” file should be set to a random, secret value that is not committed to source control, but is kept secret so that users external to the organization cannot have access to this key.

Further settings describing other case-specific deployment optimizations can be found in the official Django deployment documentation:

https://docs.djangoproject.com/en/3.1/howto/deployment/checklist/#run-manage-py-check-deploy.

Another helpful document providing an example of deployment on a Heroku server can be found at the following link:

https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Deployment

The USC TeleCARE application employs images and CSS files for content creation and styling, which are termed “static content”. These files can be found within the “telecare/static/” directory. For production deployment purposes, static content should be located in the same location for easy serving. It is important to note that Django does not serve static files itself, leaving this up to the web server. In turn, the necessary settings changes will be dependent upon the deployment setup of the web server. Settings under “telehelp/admin/settings.py’ that may need to be modified based on protocols and procedures of the hosting web server include the following:

STATIC_URL
STATIC_ROOT

For official instruction on how the “telehelp/admin/settings.py” file can be configured to match the web server setup, please refer to the following pages on how the static file management can be configured for your specific web content server.

Managing Static Files:
https://docs.djangoproject.com/en/3.1/howto/static-files/#configuring-static-files

Deploying Static Files:
https://docs.djangoproject.com/en/3.1/howto/static-files/deployment/