Ganymede Heights

HOTEL BOOKING SYSTEM | PYTHON PORTFOLIO PROTOTYPE

AUTHOR: SIAN C C THOMPSON

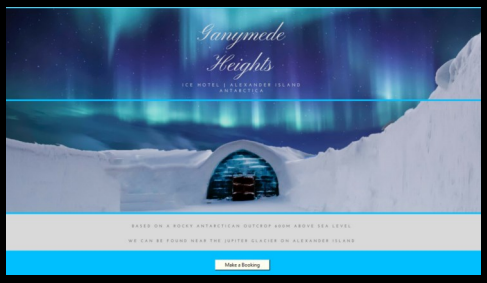
About

- Example of front and back-end development with an integrated Mongo DB database
- The application simulates a user-browser experience from the point of homepage discovery for a fictional ice hotel website based in Antarctica

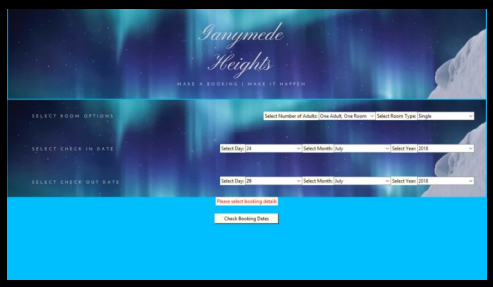
Project Modules and Tools

Preview App Images

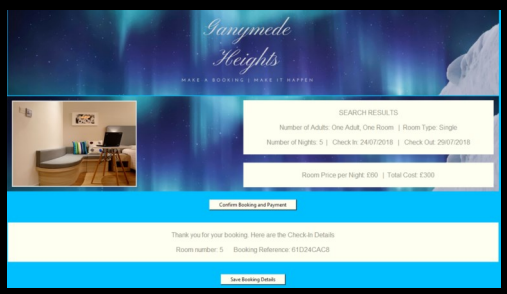
1) Hotel Homepage



2) Hotel Booking Page



3) Room Details Page



***Please see Ganymede Heights Adobe XDCC Demo here: https://adobe.ly/2LkmfkB**

Neither MongoDB or Python are required to view this

User Experience & Back-end

- The online publishing tool **Canva** was used for creating to visually appealing banners to standard size
- Clicking the 'Make a Booking' button takes the user to the booking page
- On the booking page, the user can select various booking choices
- Clicking 'Check Booking Dates' prompts various checks e.g. if the booking end date is earlier than the start date, the warning box message will change
- On the room details page, the user can review booking choices and proceed to press 'Confirm Booking and Payment'
- This prompts a new message box with the room number and booking reference. The user can also save a png image of the booking

Key Learnings in Python

- Challenge: Overcome 'flash ' between window transitions
- Solution: 'after ' and 'wm_withdraw' methods helped create a smooth 50 ms transition as a root window is closed and another is opened
- Challenge: Creating margins around widgets arranged over a background image
- Solution: a) 'pack' a label widget just containing '\n' for new lines, b) 'place' a label with the background image , c) 'pack' remaining widgets
- Challenge: Capturing a printable png image of the booking
- Solution: A 'wm_state('zoomed')' status for the window enabled a full-screen view. Whilst 'ImageGrab' can helps save a screenshot with booking details, the entire screen will only be captured if display scaling for high DPI settings are disabled for Python. However this has a negative impact on the GUI layout highlighting limitations of using tkinter widgets.

Ganymede Heights

HOTEL BOOKING SYSTEM | PYTHON PORTFOLIO PROTOTYPE

AUTHOR: SIAN C C THOMPSON

Ganymede Heights Python Project Set-Up

- **GitHub Project Download Link:** https://github.com/Kremzeeg/GanymedeHeightsHotelBookingProject
- Module pre-requisites are in requirements.txt
- Opening the project in PyCharm should prompt for modules from the requirements.txt to be installed
- Alternatively use pip install requirements.txt in the command line

MongoDB Set-Up

- Mongo DB Community Edition 4.0 was installed for this project
- Manual for installing MongoDB on Linux, macOS and Windows is available here:
- https://docs.mongodb.com/manual/administration/install-community/

Running the Application

1) Initialising the MongoDB Database

- Once MongoDB has installed open the command line and type **mongod**
- This initiates the Mongo DB database which will 'listen' for commands run from the python application or mongo in the command line
- Prior to running the python application for the first time, enter the following steps in a separate command line:

1) mongo 2) use HotelBookingDB 3) db.createCollection('HotelBookings')

- The 'HotelBookings' collection needs to be created prior to running the python application as it will query any precious bookings in the collection

2) Initialising the Python App

- The application can be run from the hotelProjectMain.py file
- Please note that the JSON for the hotel booking is saved to the database at the point that the 'Confirm Booking and Button' payment is pressed**. All bookings saved in the database will be displayed in the shell
- When the 'Save Booking Details' button is pressed a hotelBooking.png file is saved to the project root folder when the 'Save Booking Details' button is pressed. This returns a screenshot of the page

3) Quick Database queries

- In the command line any accumulated hotel bookings can be removed in mongo using ' db.HotelBookings.drop()'. However, remember the collection would have to be initiated again in order for the python application to work
- Execute db.HotelBookings.find({}) in the command line to view all HotelBooking JSON files
- Alternative pymongo methods are available in the HotelBookingDB.py file

****Disclaimer:** There is no integrated Flask module for an online database development environment. The application does not harvest any personally identifiable information nor does it process any real-life payment transactions.