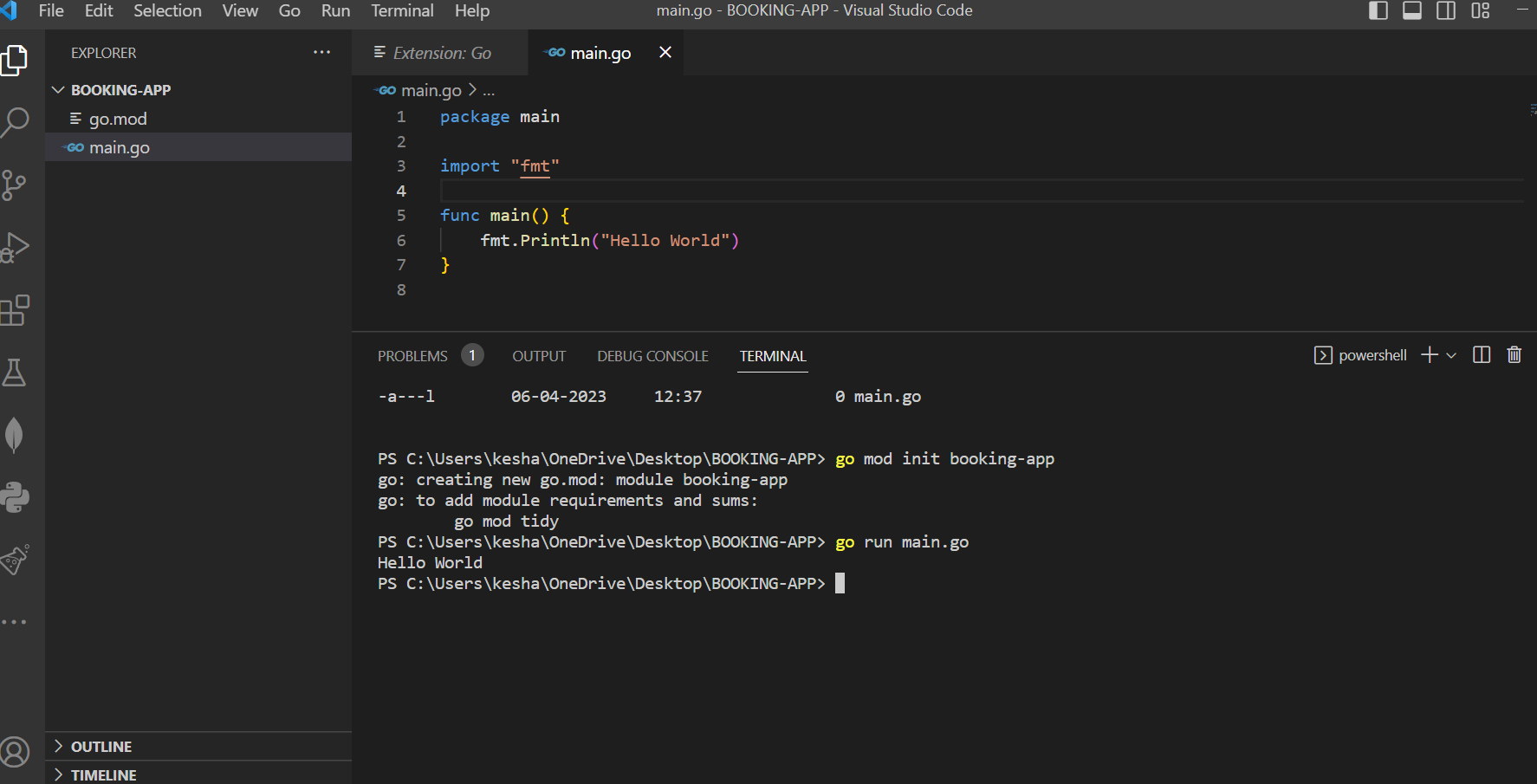
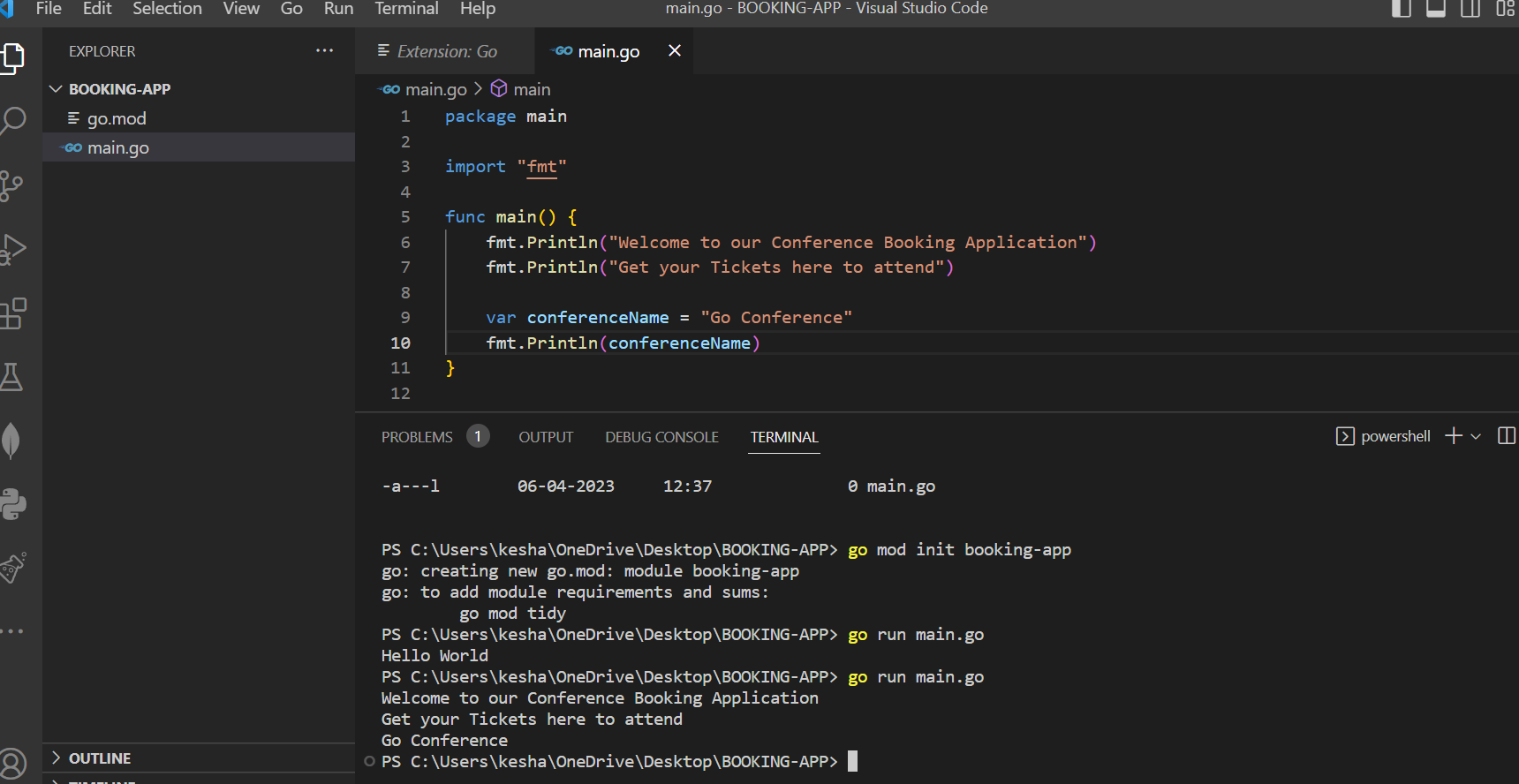
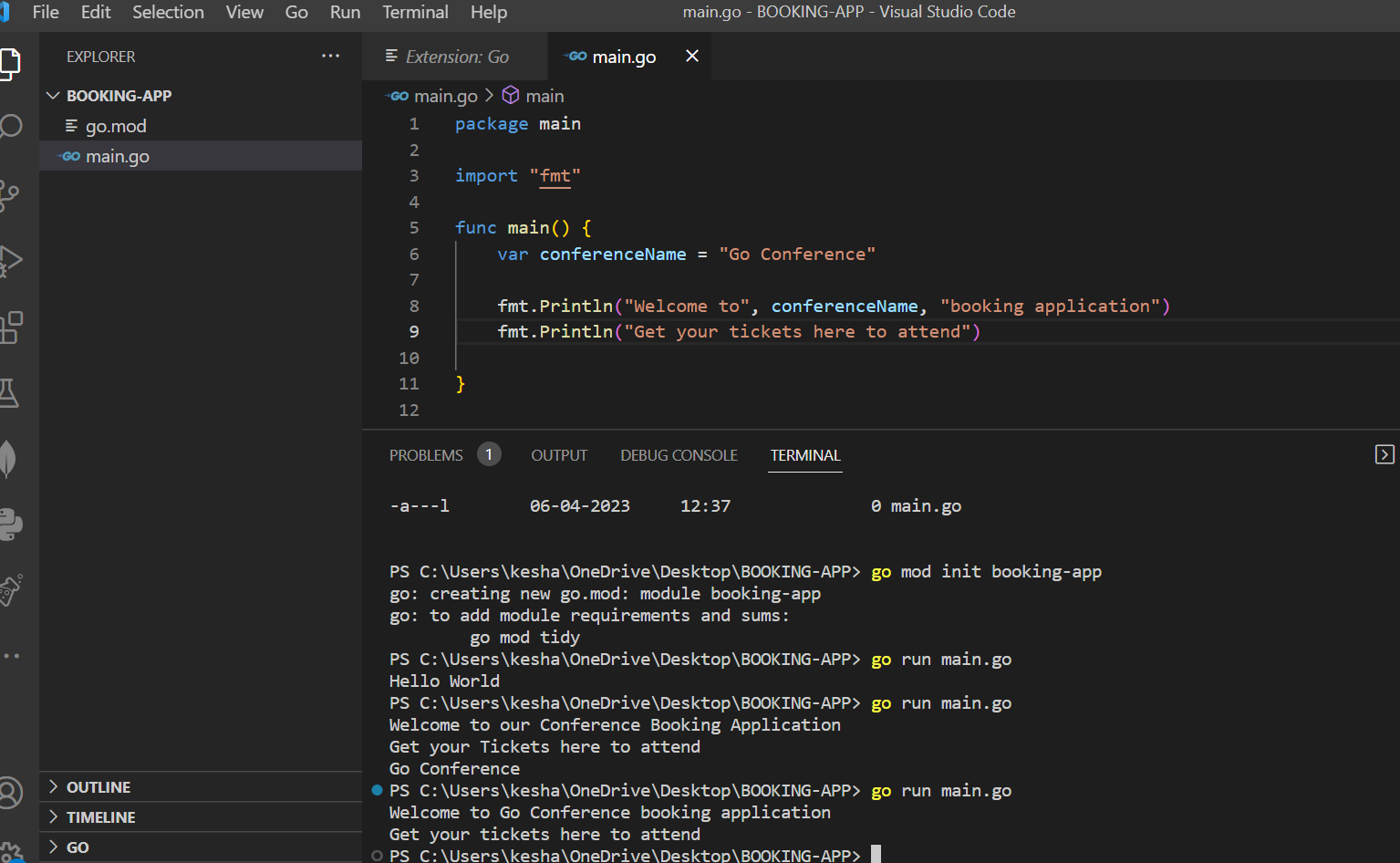
GOLANG ASSESSMENT 1

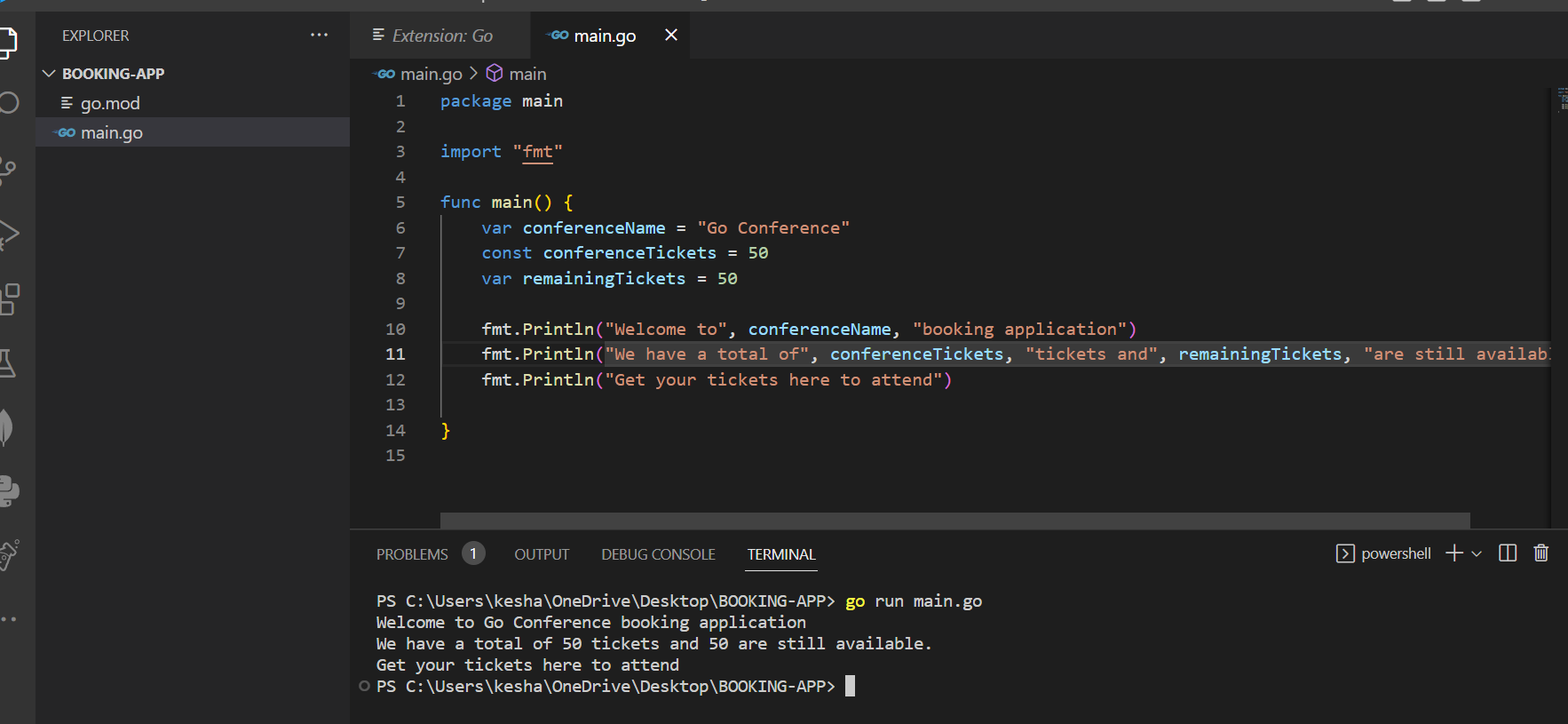
Booking App

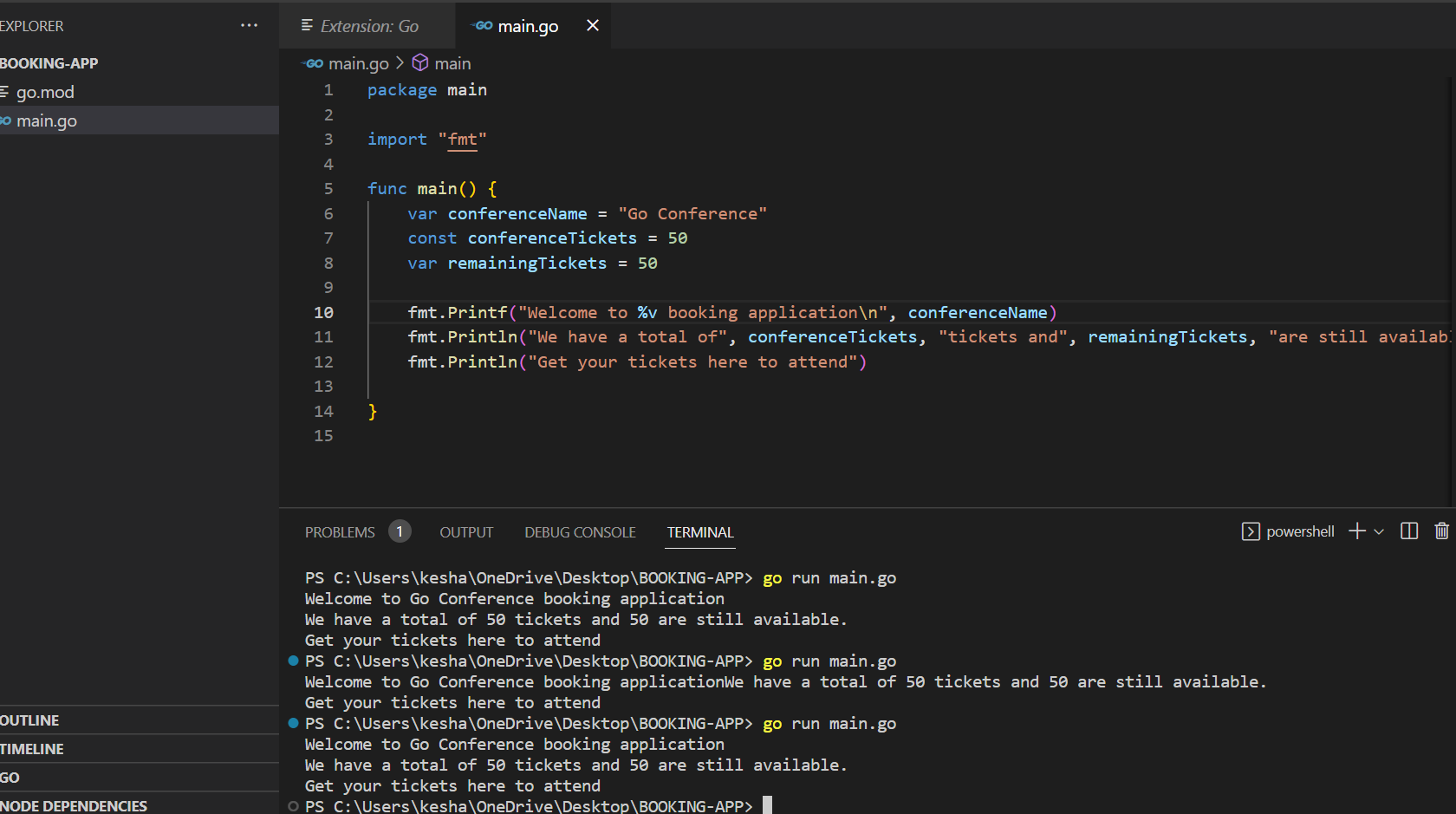
Variables, Data Types

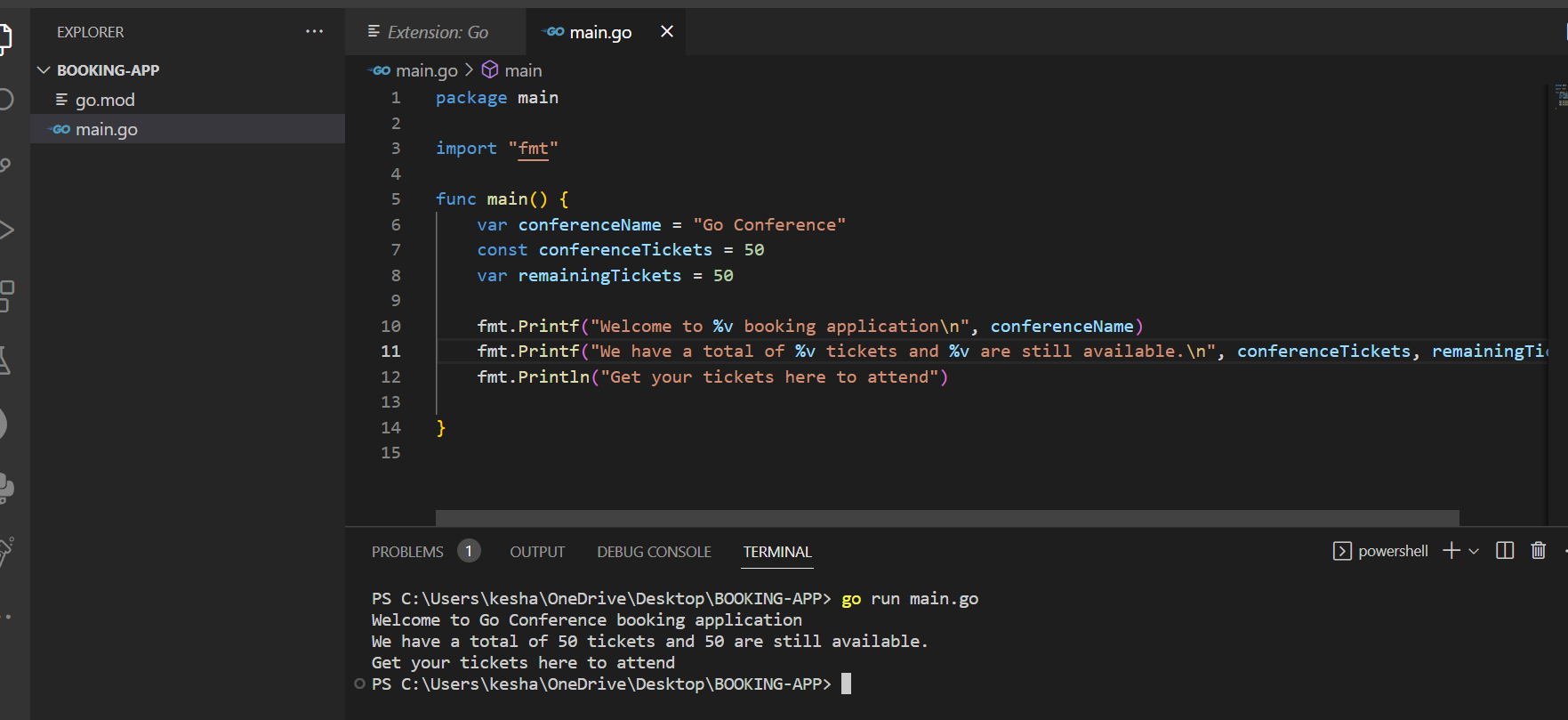




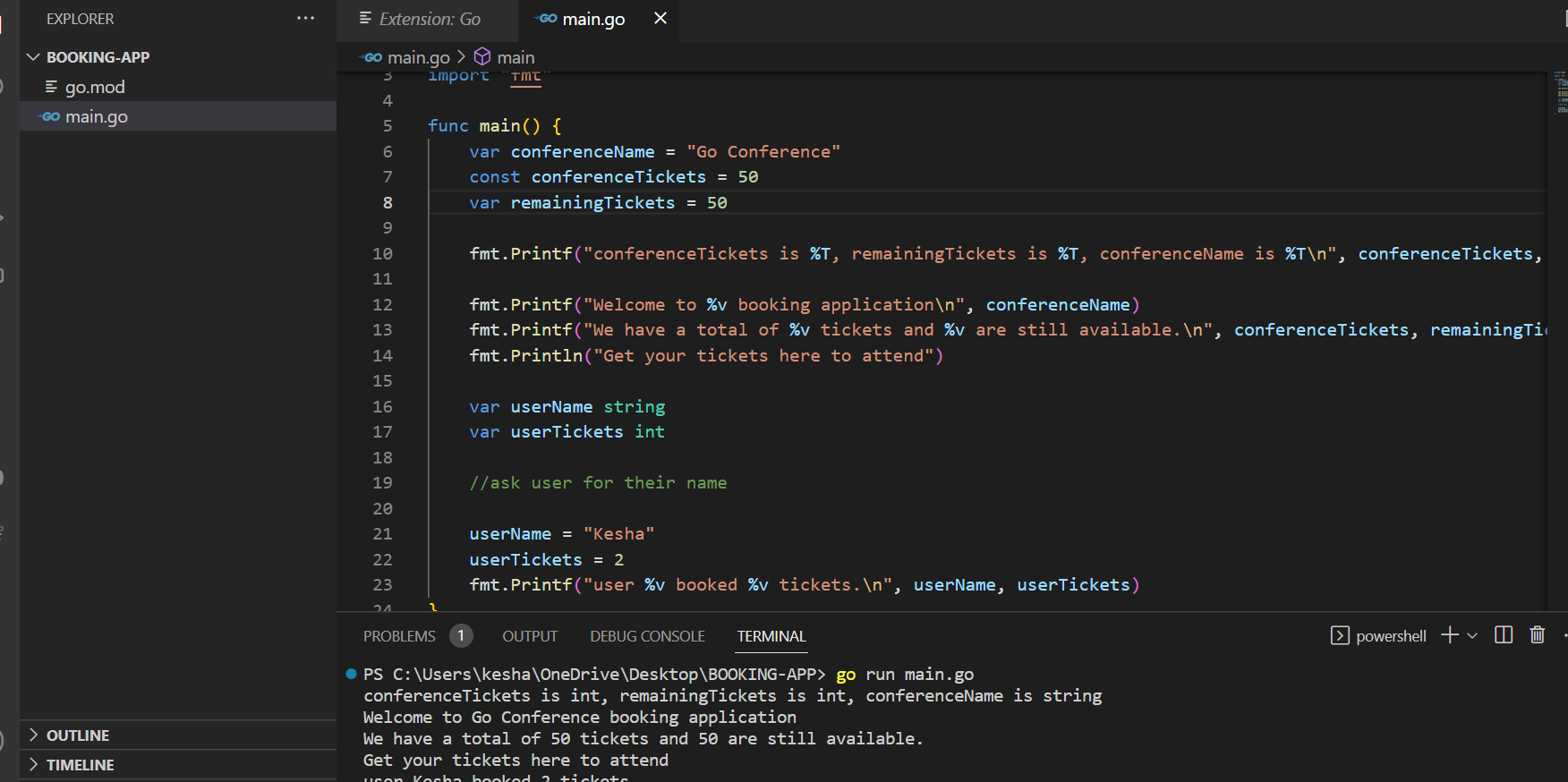


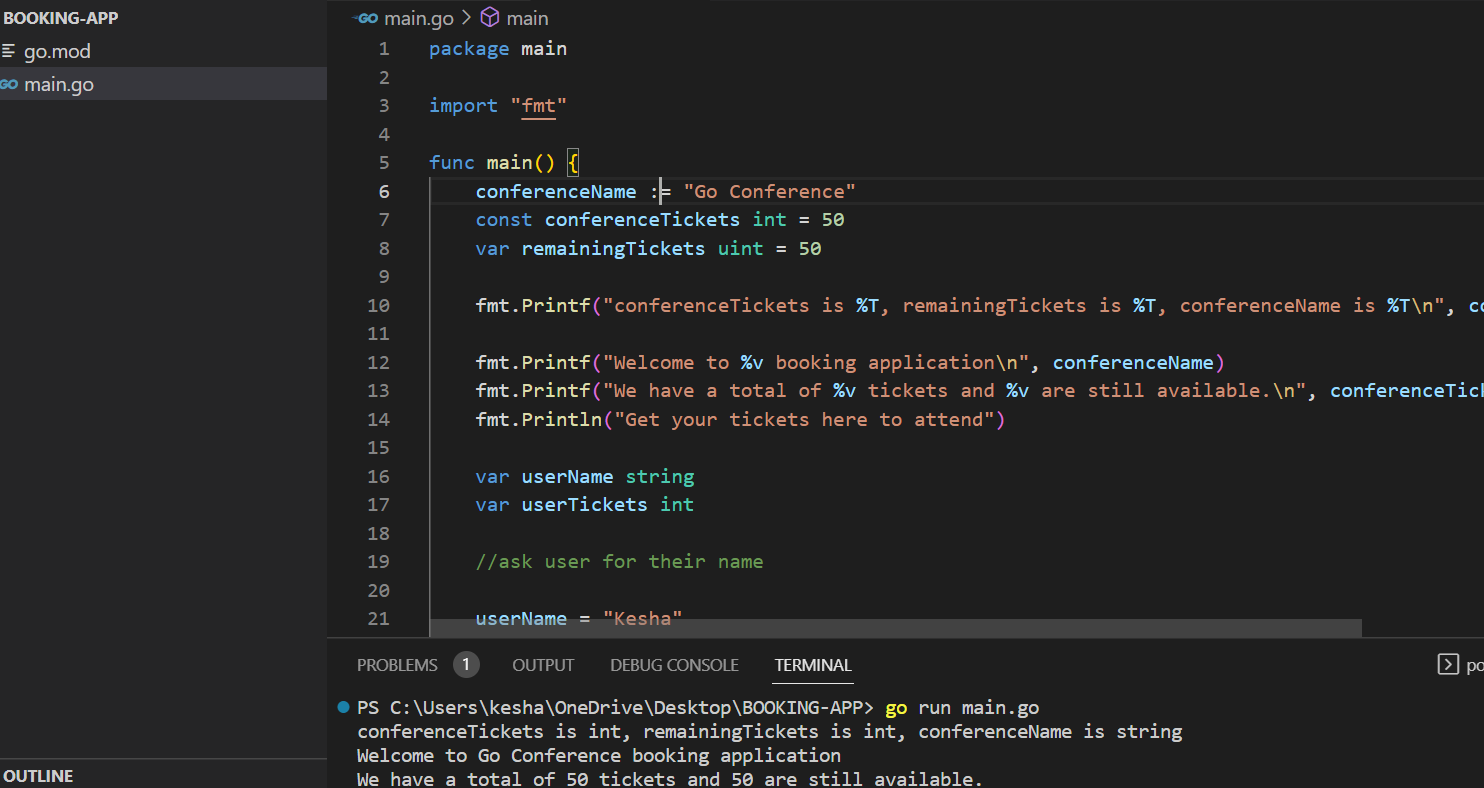




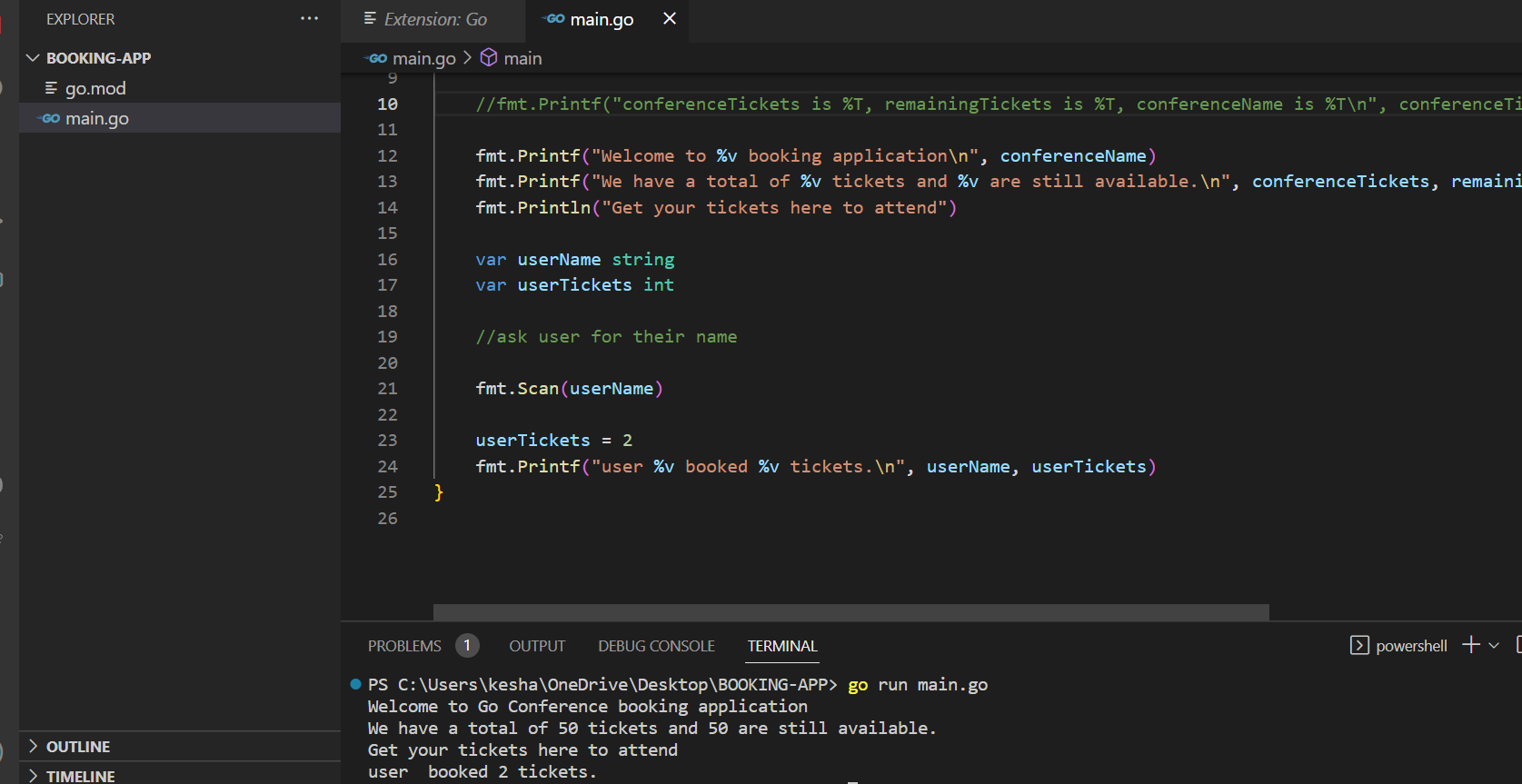


Data Types

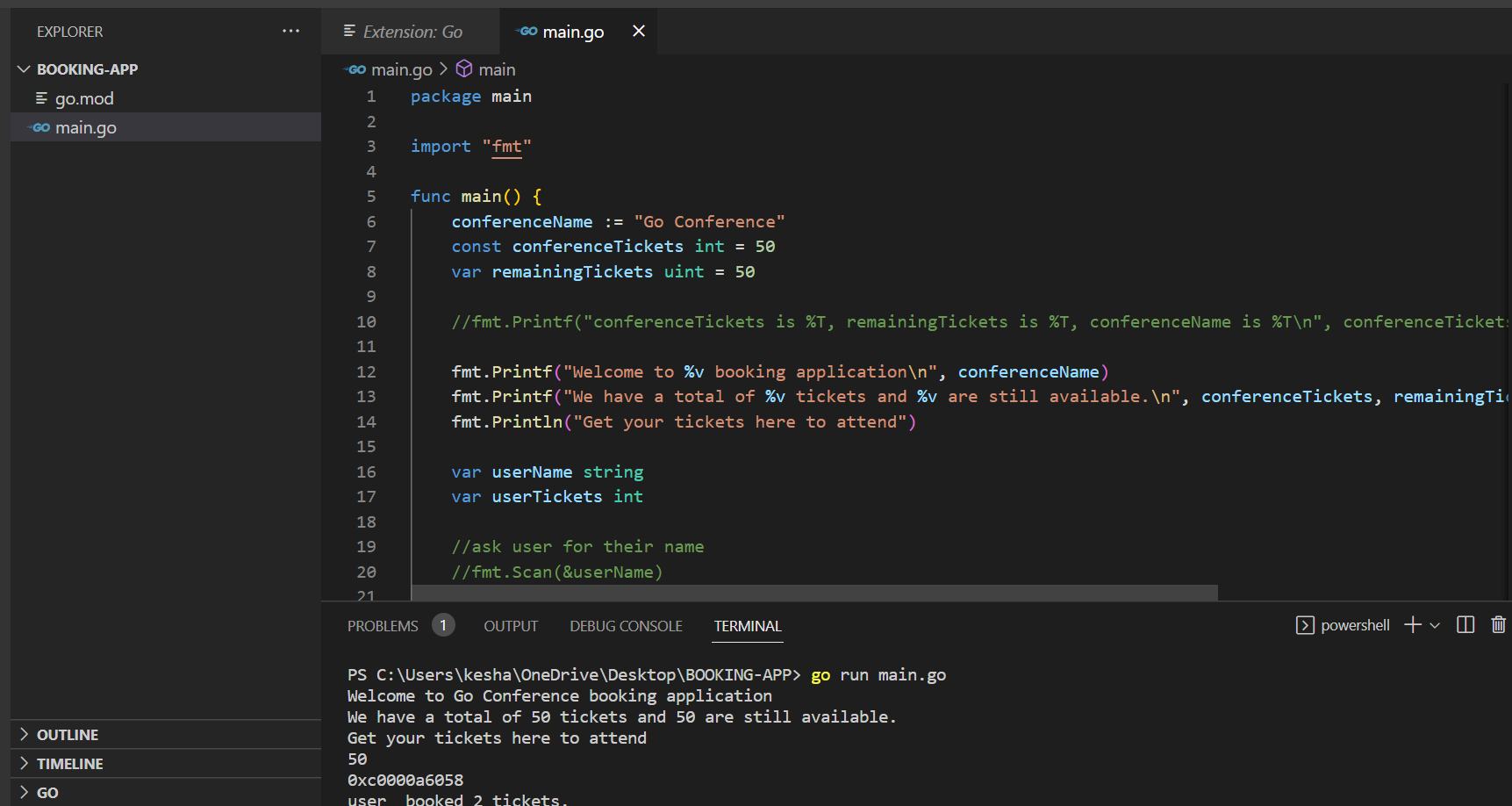


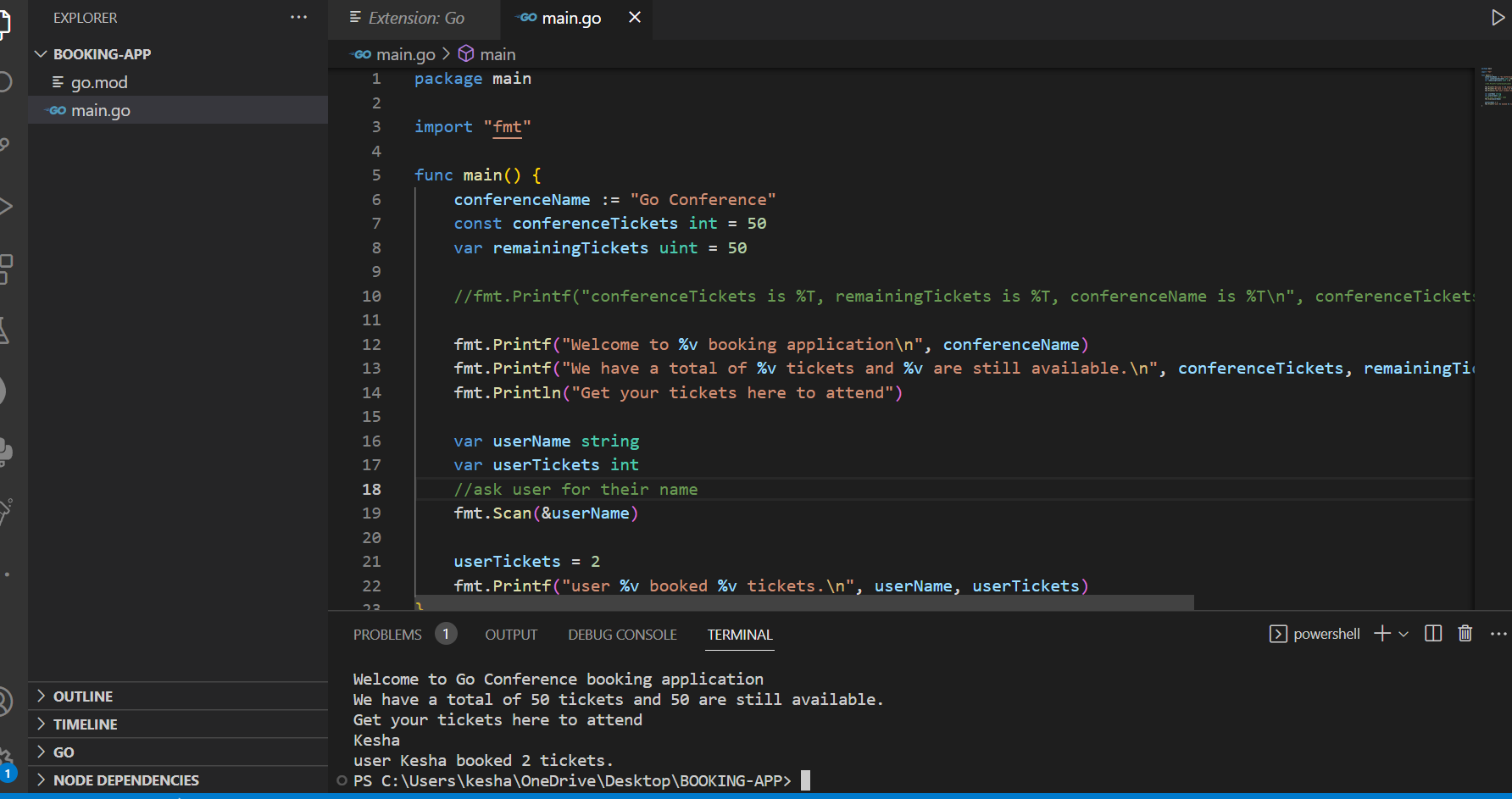


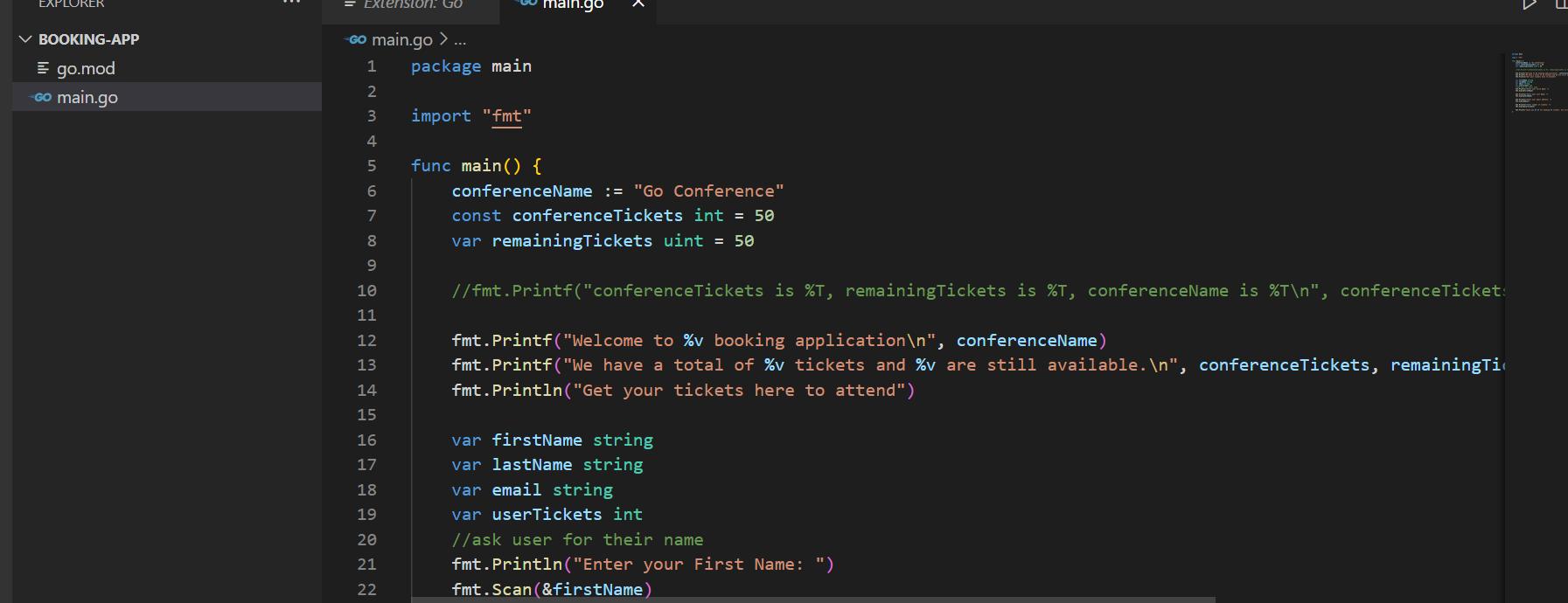
Getting user input

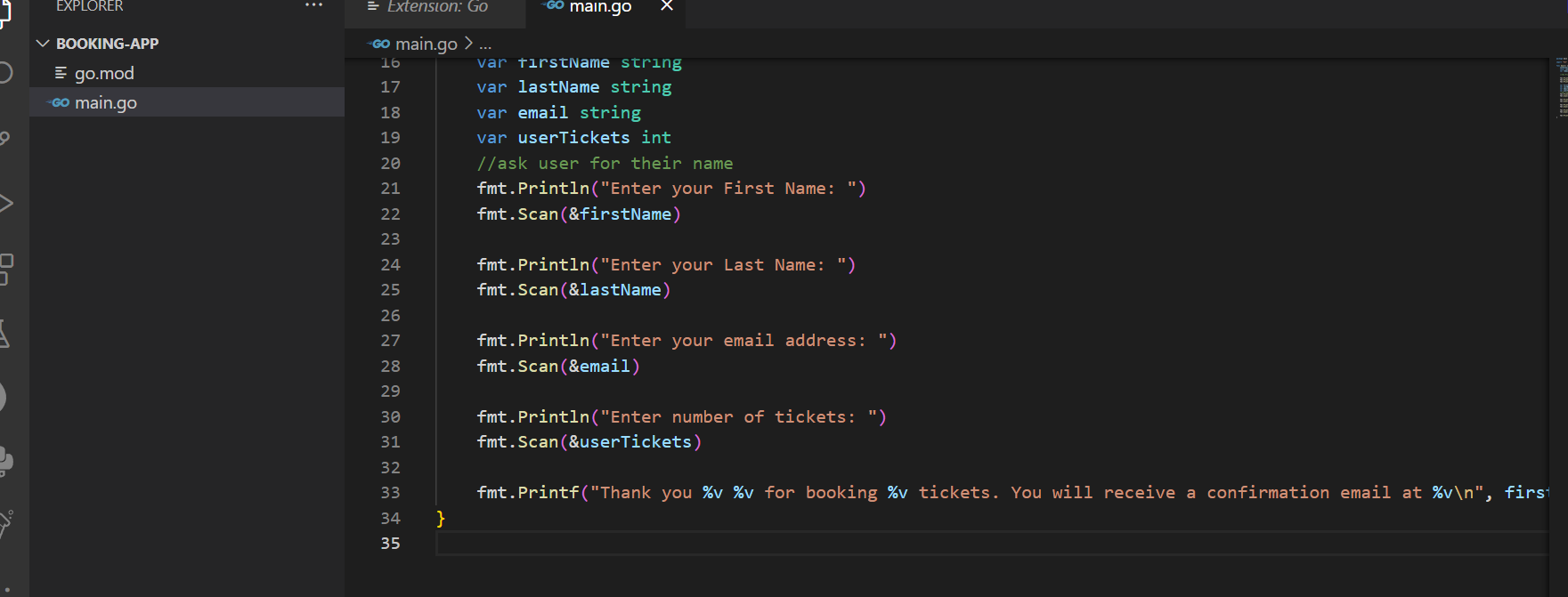


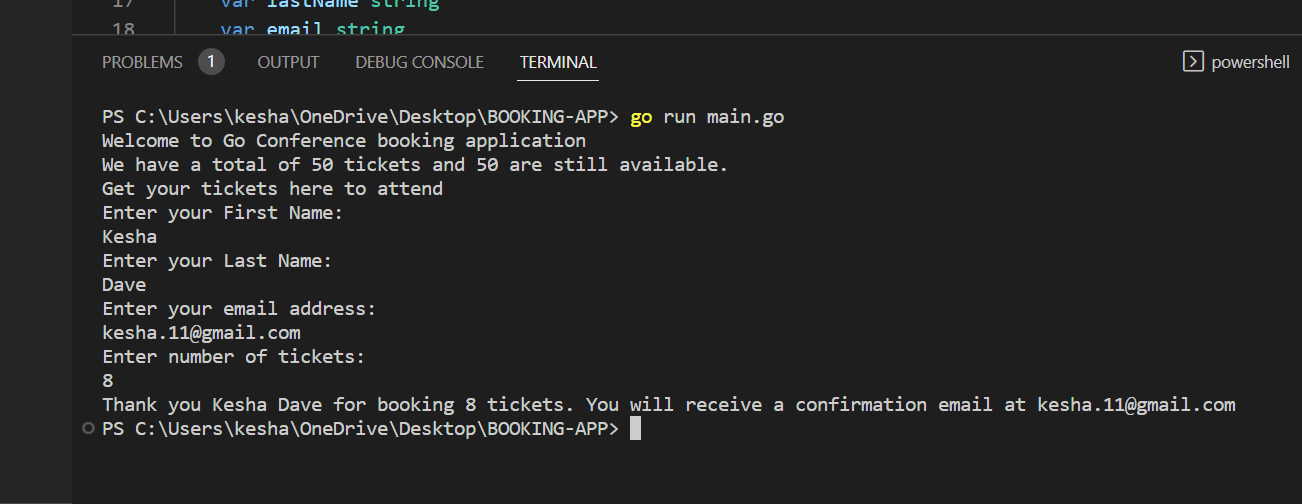
Pointers

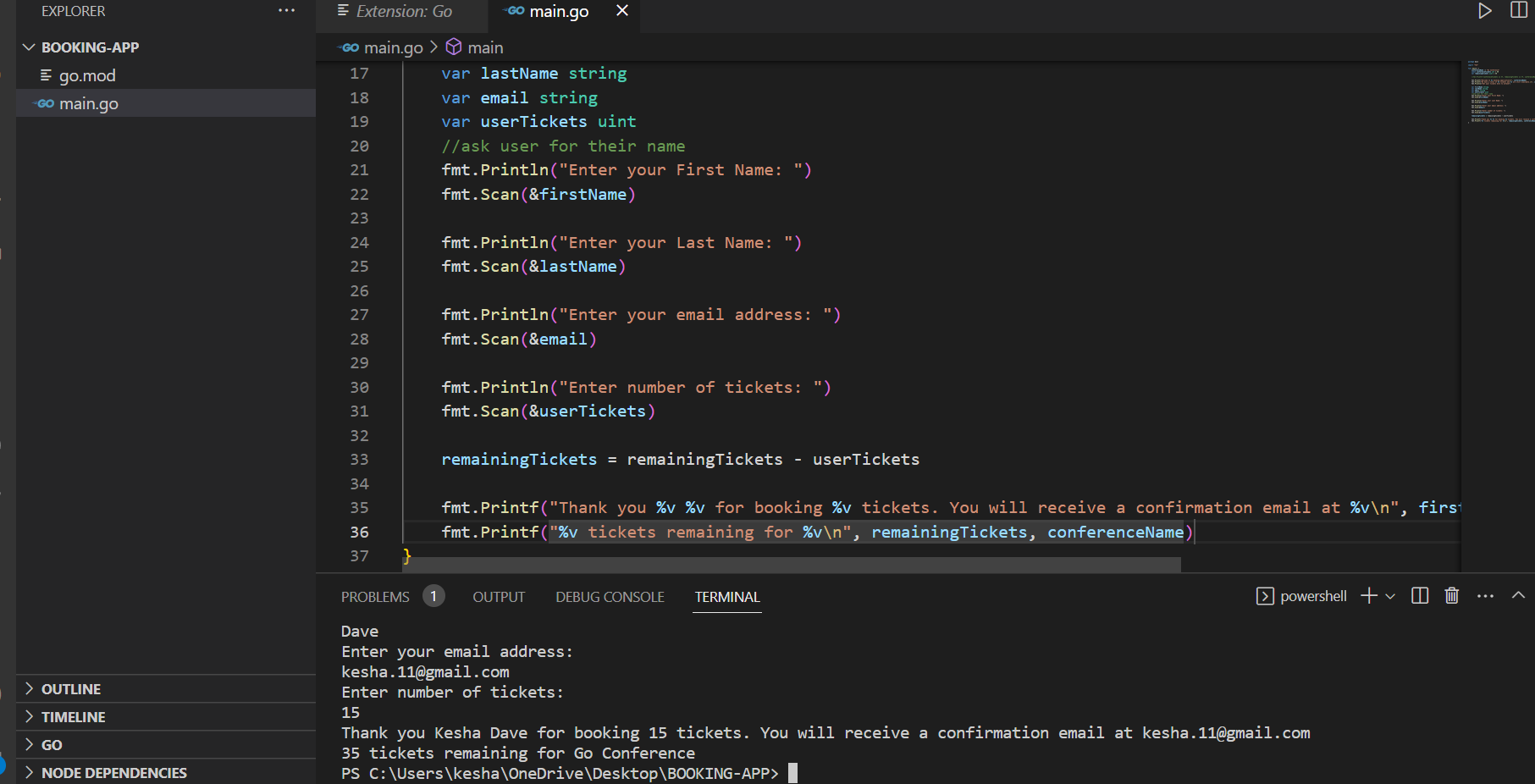




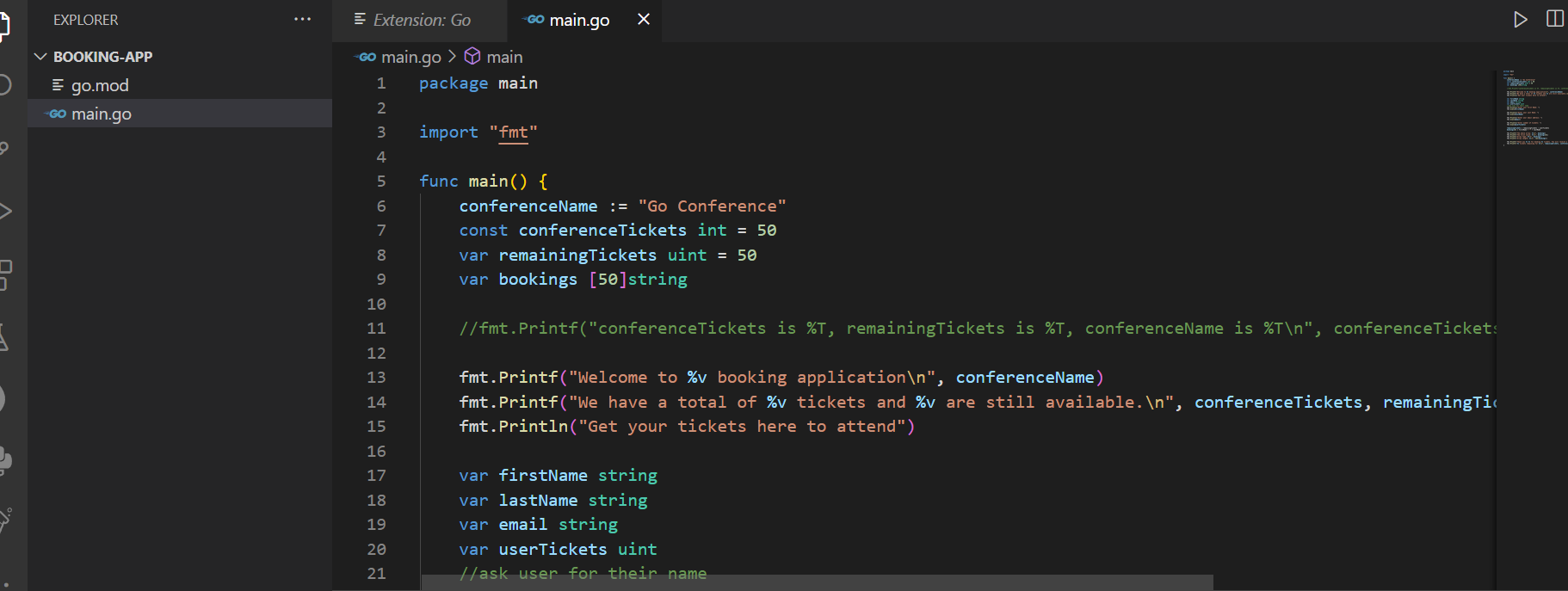


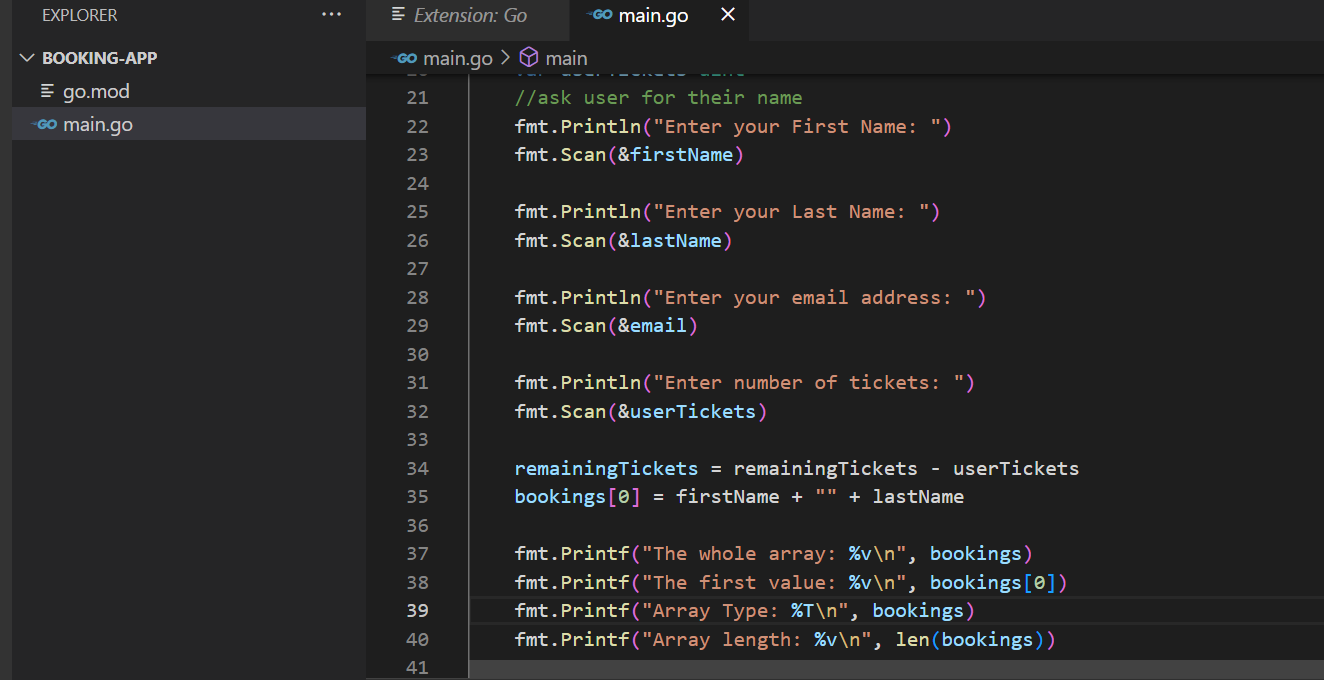


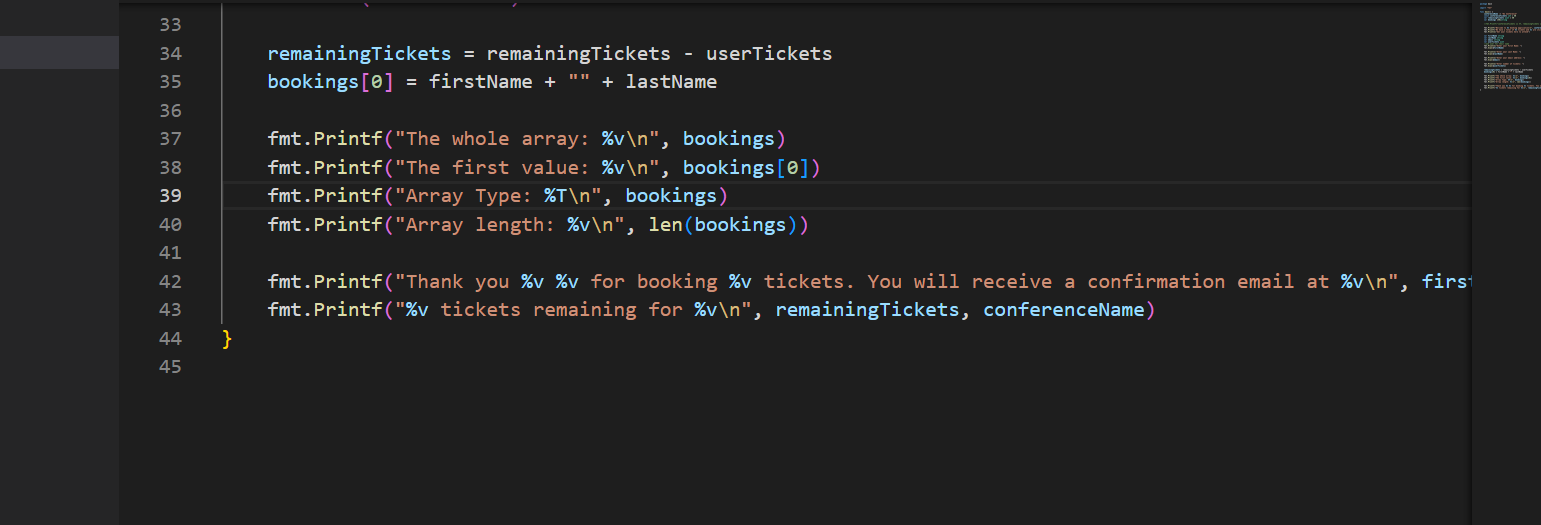


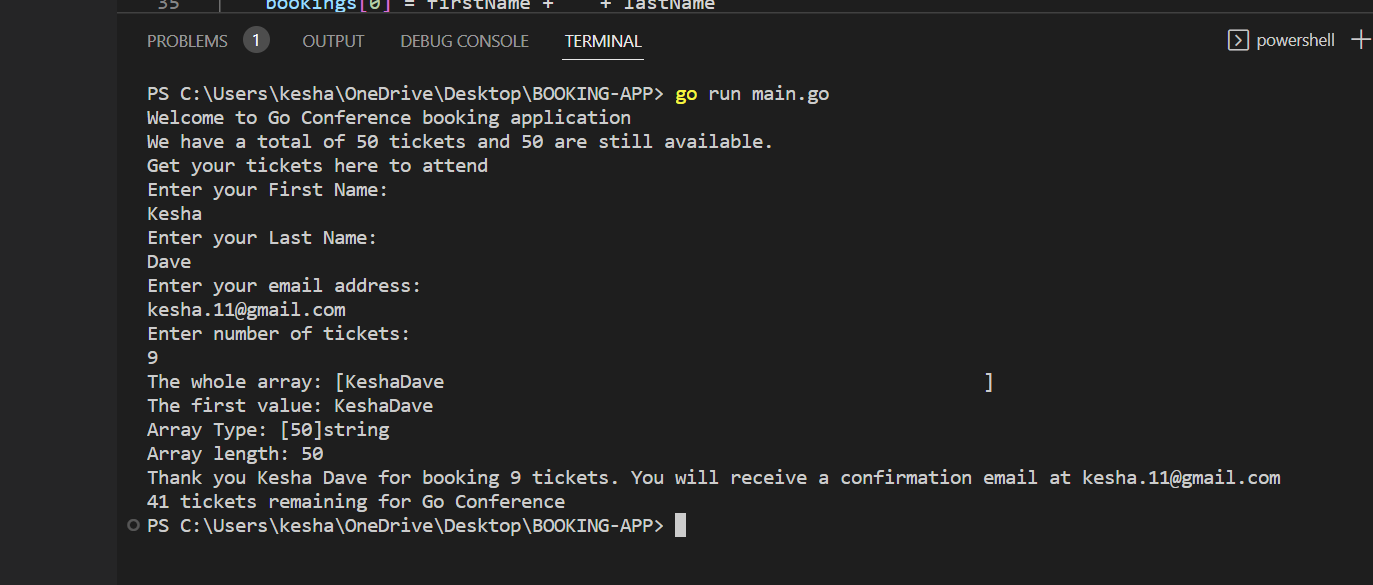


Arrays

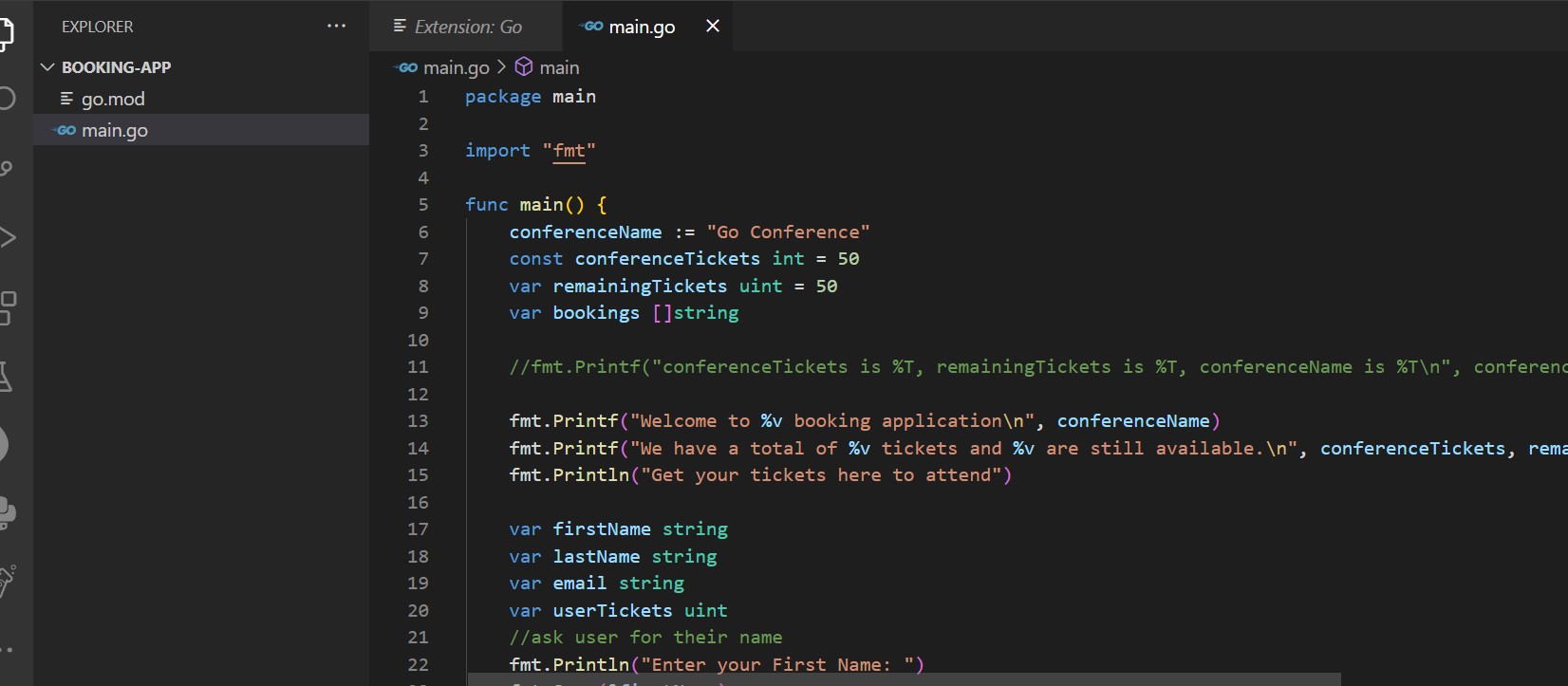


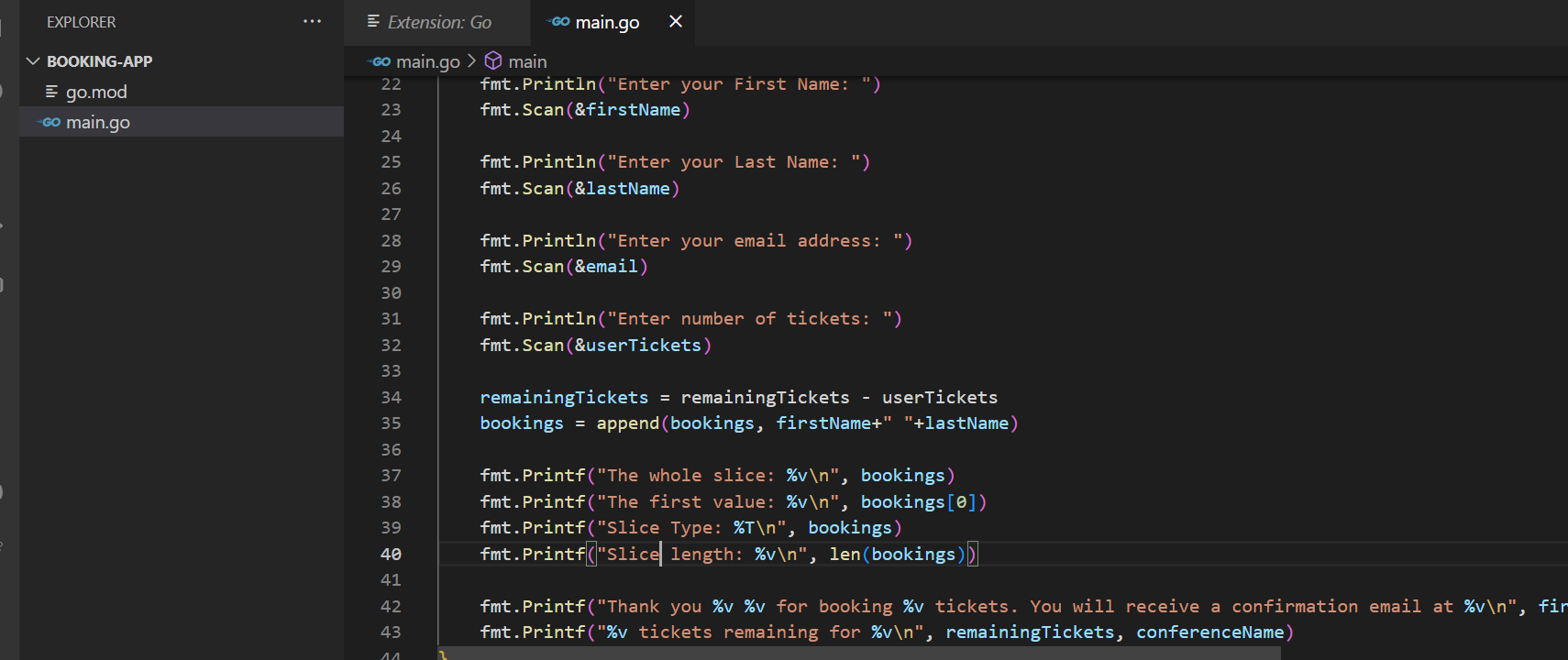


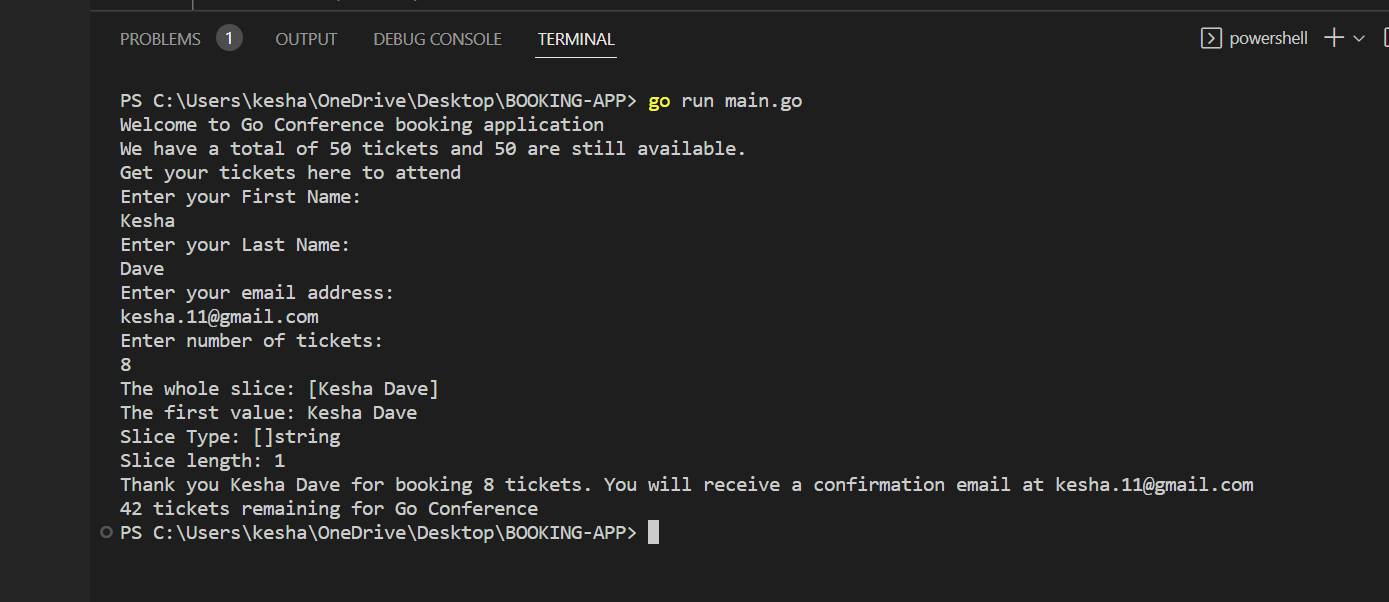


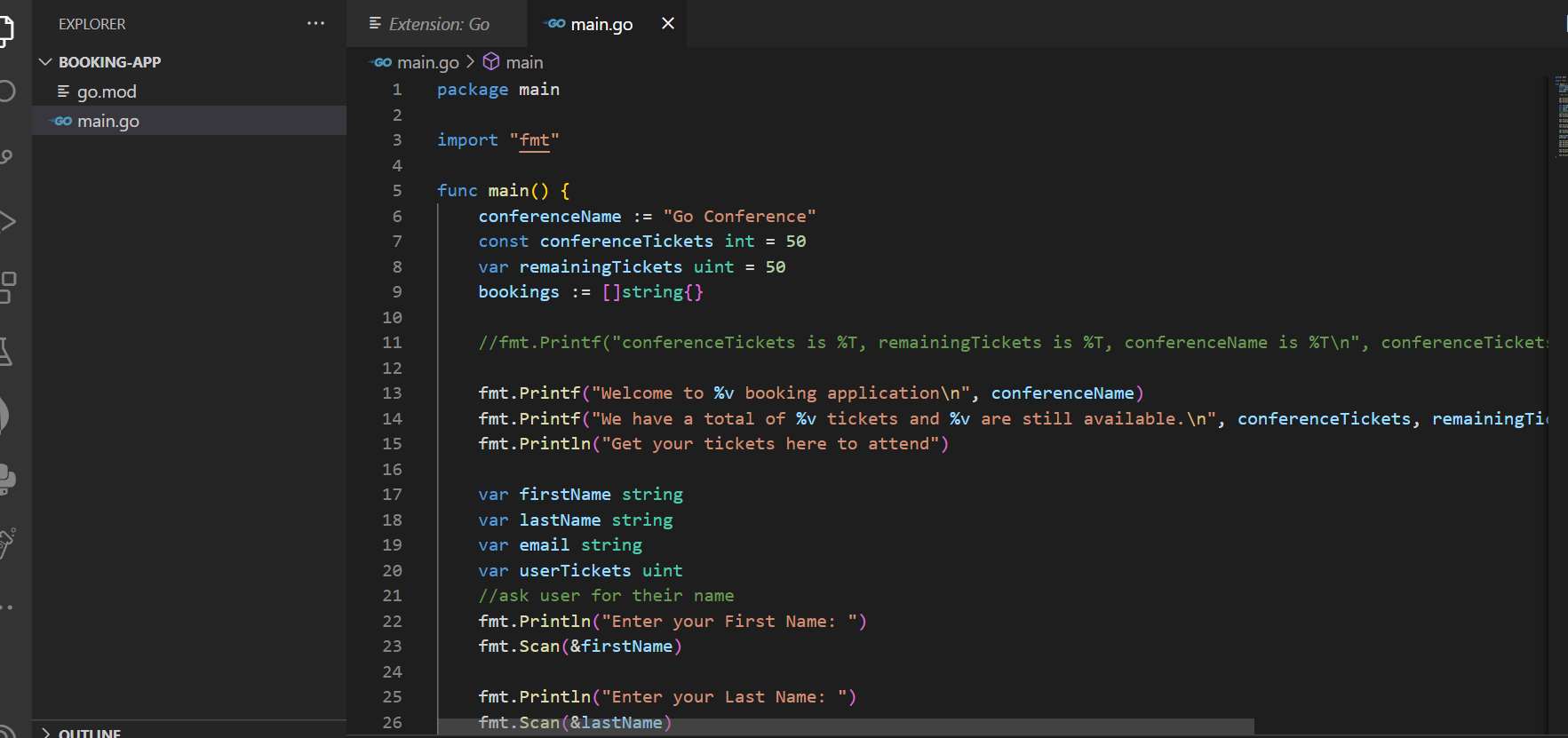


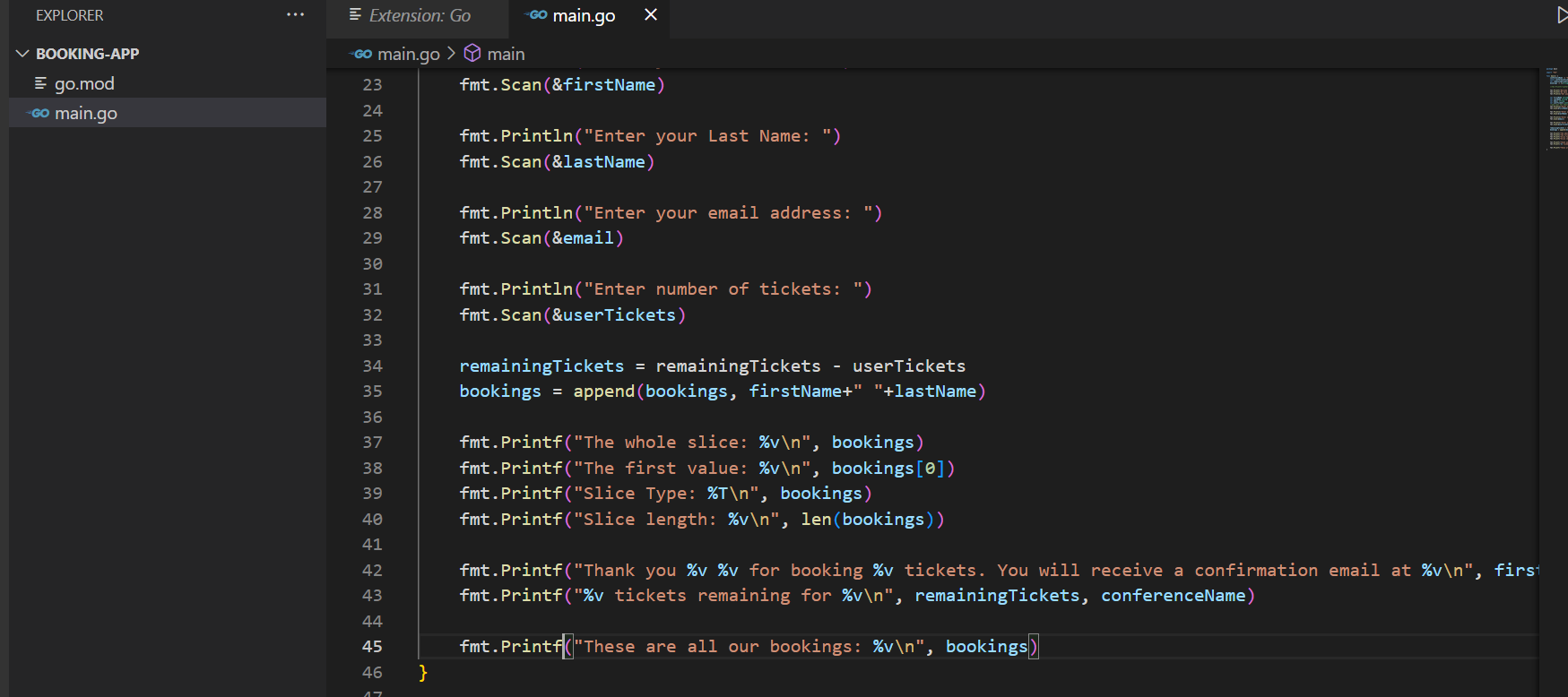
Slices

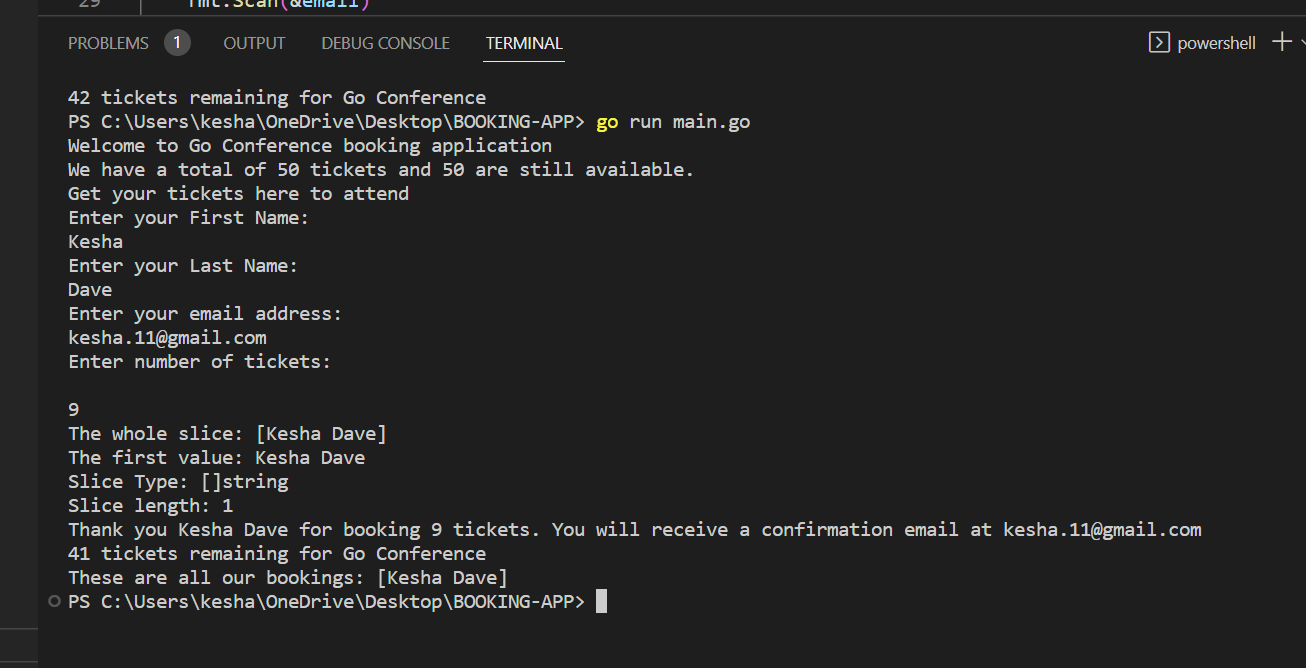












LOOPS IN GO (Remaining from here)

* Leetcode Problems with Golang

The problem states that you may not slant the container. In the image, you can see that the answer involves two sides where one is height 8, and the other is height 7. This is important to note, because we must make sure that the sides of the rectangle are balanced, so when dealing with two uneven sides, we will perform the computation for rectangle area with whichever side is the smaller of the two.

The difference between the indexes j and i will provide us the length of the rectangle. So our computation for the rectangle’s area will look like:

area = rectLength \* rectHeight

rectLength = j-i

rectHeight= height[i]

* **Naïve approach**

func maxArea(height []int) int {

var maxArea int = 0

var size int = len(height)

for i := 0; i < size; i++ {

currentArea := 0

for j := 0; j < size; j++ {

segmentLength := j-i

var segmentHeight int = height[i]

if(segmentHeight >= height[j]) {

segmentHeight = height[j]

}

currentArea = segmentLength\*segmentHeight

if(currentArea > maxArea) {

maxArea = currentArea

}

}

}

return maxArea

}

* **Optimized Solution**

func maxArea(height []int) int {

var maxArea int = 0

var i int = 0

var j int = len(height)-1

var result int = 0

for i < j {

if(height[i] <= height[j]) {

result = height[i] \* (j-i)

i++

} else {

result = height[j] \* (j-i)

j--

}

if(result > maxArea) {

maxArea = result

}

}

return maxArea

}