Digital Turbine – Developer Test

The purpose of this test is to establish how you approach problems and what programming methodologies you use.

When you are doing this test, please aim to produce code that is using best practices and standards. Also, any performance enhancements you can add to this code to make the solution faster should be included. Feel free to use any resources, so long as you are comfortable that if challenged, you know what you have done and why.

The project you will be creating will be a simple C# ASP.Net MVC web site. You must use razor to build your pages. Below are the two code stubs that will need to be added to the project. Also feel free to change any of the supplied code to fit your needs, so long as the final solution meets the requirements. There must be 10000 people in the list.

There are a number of questions I would like you to address following the code stubs.

Person Class:

```
class Person {
        public string Name { get; set; }
        public int Age { get; set; }
        public override string ToString() {
                return "My name is '" + Name + "' and I am " + Age + " years old.";
        }
}
```

Init Stub:

```
public static List<Person> InitPeople() {
        List<Person> people = new List<Person>();
        Random rnd = new Random();
        for (int i = 0; i < 10000; i++) {
                people.Add(new Person() {
                        Name = "Person #" + i.ToString(),
                        Age = rnd.Next(1, 99)
                });
        }
        return people;
}
```

Tasks:

1. Update the class structure, to implement the following races: Angles, Saxons, Jutes and Asians
2. Implement a way of getting the height for a person based on the following logic:

   For Angles and Saxons, Height = (1.5 + (Age * 0.45))

   For Jutes, Height = ((age * 1.6) / 2)

   For Asians, Height = (1.7 + ((age + 2) * 0.23))
3. Update the loop that creates people, so that it randomly creates different races of person.
4. Write a loop to add 1 year to each person’s age
5. Build a webpage that shows a drop down list of races. When the user selects a race, the page should reload to show all the people, in a list or table, in that race whose age is even, ordered by age. Show their name, age and height.
6. Build a basic API endpoint that returns a JSON document with the following pieces of information, based on the list of people:
   a. Number of people
   b. Average, Min and Max Age
   c. Count of the number of people in each race