**1. https://github.com/rvsp/typescript-oops/blob/master/Practice/Movie.md**

// 1. https://github.com/rvsp/typescript-oops/blob/master/Practice/Movie.md

// defining the class movie in the we are writing the required arguments like title, studio, and rating = PG default

class Movie {

constructor(title, studio, rating = "PG") {

this.title = title;

this.studio = studio;

this.rating = rating;

}

}

// creating the function object with required arguments.

function create\_object(title, studio, rating) {

return new Movie(title, studio, rating);

}

// getting the list with whose rating is PG so we are getting that output.

function getPG(list) {

return list.filter((film) => film.rating === "PG");

}

// passing the constructor objects to the class constructor.

let film1 = new Movie("Manam", "Annapurna Studios", "G");

let film2 = new Movie("Drushyam-2", "AmazonPrime");

let film3 = new Movie("F2", "Ramanaidu Studios", "PG13");

let film4 = new Movie("Adbutham", "Padmalaya Studios");

let film5 = new Movie("Aranyaa", "Ramoji Film City");

// getting the required outputs after the calling of them.

console.log("\nBelow is the output for the question 1(a)\n");

console.log(film1);

console.log("\nBelow is the output for the question 1(b)\n");

console.log(film2);

console.log("\nBelow is the output for the question 1(c)\n");

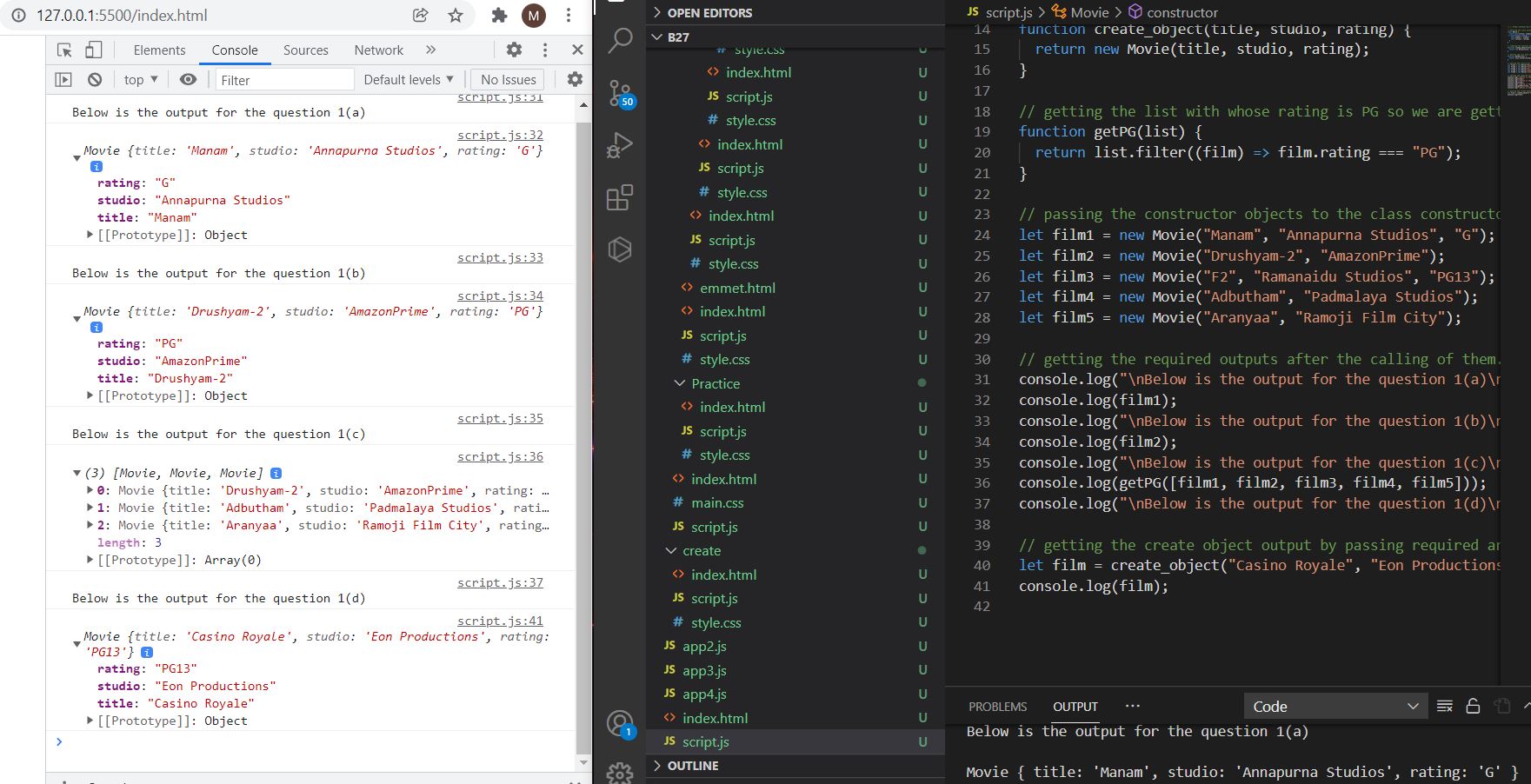
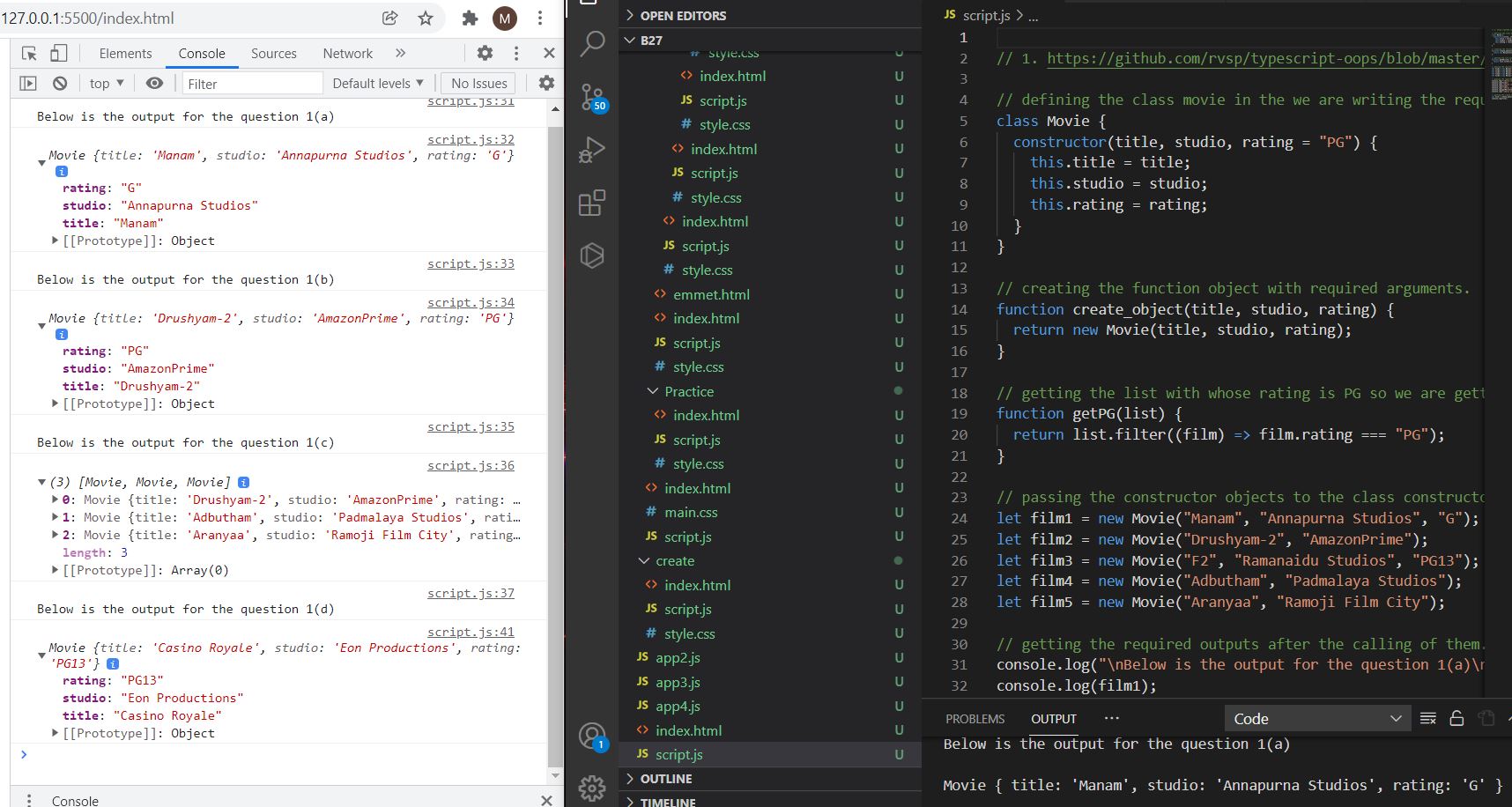
console.log(getPG([film1, film2, film3, film4, film5]));

console.log("\nBelow is the output for the question 1(d)\n");

// getting the create object output by passing required arguments values.

let film = create\_object("Casino Royale", "Eon Productions", "PG13");

console.log(film);



**2. https://github.com/rvsp/typescript-oops/blob/master/Practice/class-circle.md**

// 2. https://github.com/rvsp/typescript-oops/blob/master/Practice/class-circle.md

// Creating the class constructor with the radius and the color attributes.

class Circle {

constructor(radius, color) {

this.radius = radius;

this.color = color;

}

// writing the getradius, setradius, getcolor, toString and getArea and getCircumference methods.

getRadius() {

return this.radius;

}

setRadius(radius) {

this.radius = radius;

}

getColor() {

return this.color;

}

setColor(color) {

this.color = color;

}

toString() {

return `Circle [radius=${this.radius},color=${this.color}]`;

}

getArea() {

// returning the area by using the pi \* r \* r formula

return Math.PI \* Math.pow(this.radius, 2);

}

getCircumference() {

// returning the circumference by using the 2\* pi\* r formula.

return 2 \* Math.PI \* this.radius;

}

}

// calling the object by using the new constrctor and passing the values.

let c1 = new Circle(Math.ceil(Math.random() \* 5), "blue");

// setting the radius to the the 2.5

c1.setRadius(2.5);

// getting the output using console.log()

console.log(c1.getRadius());

// setting thr color to the red.

c1.setColor("red");

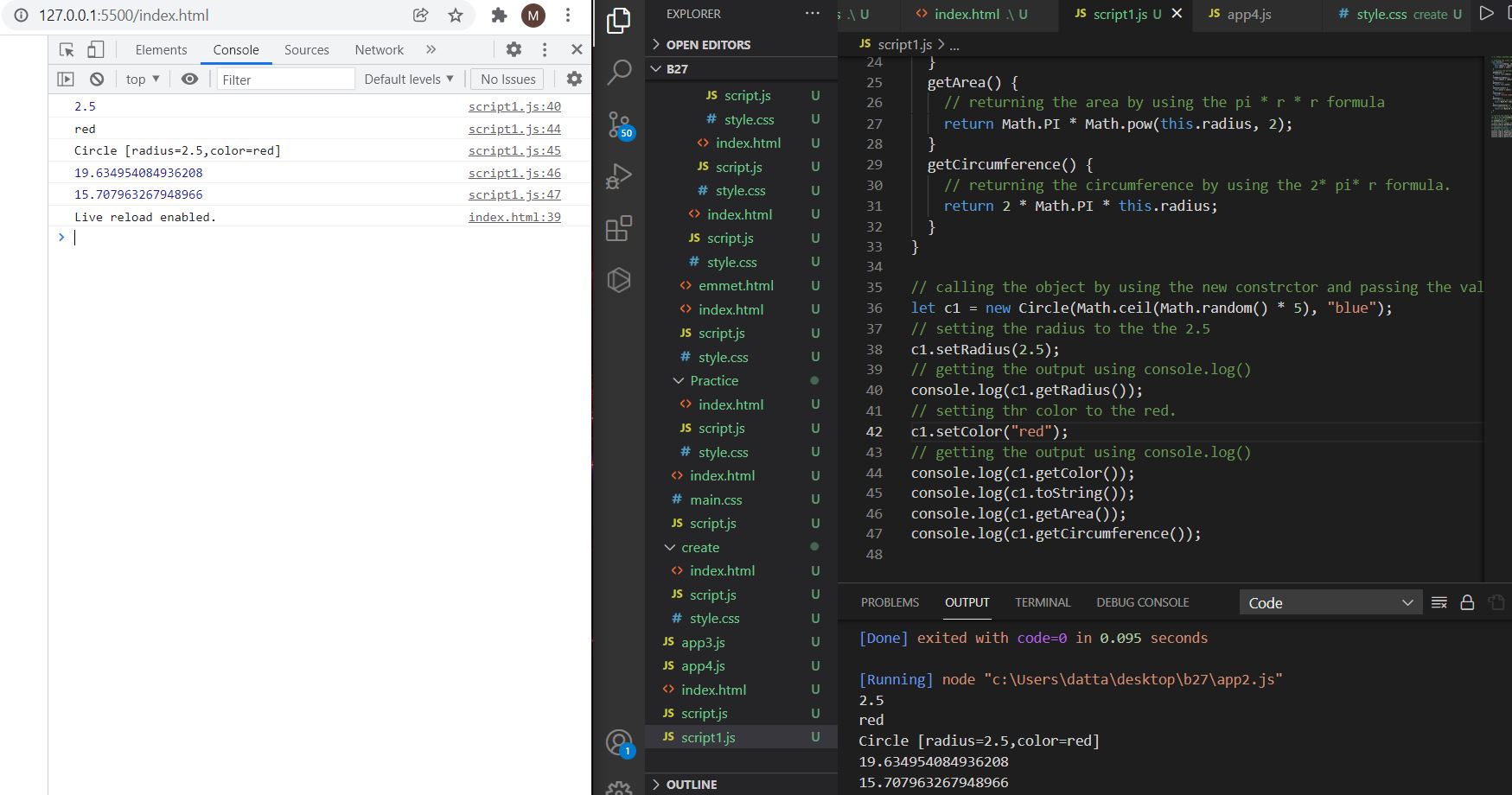
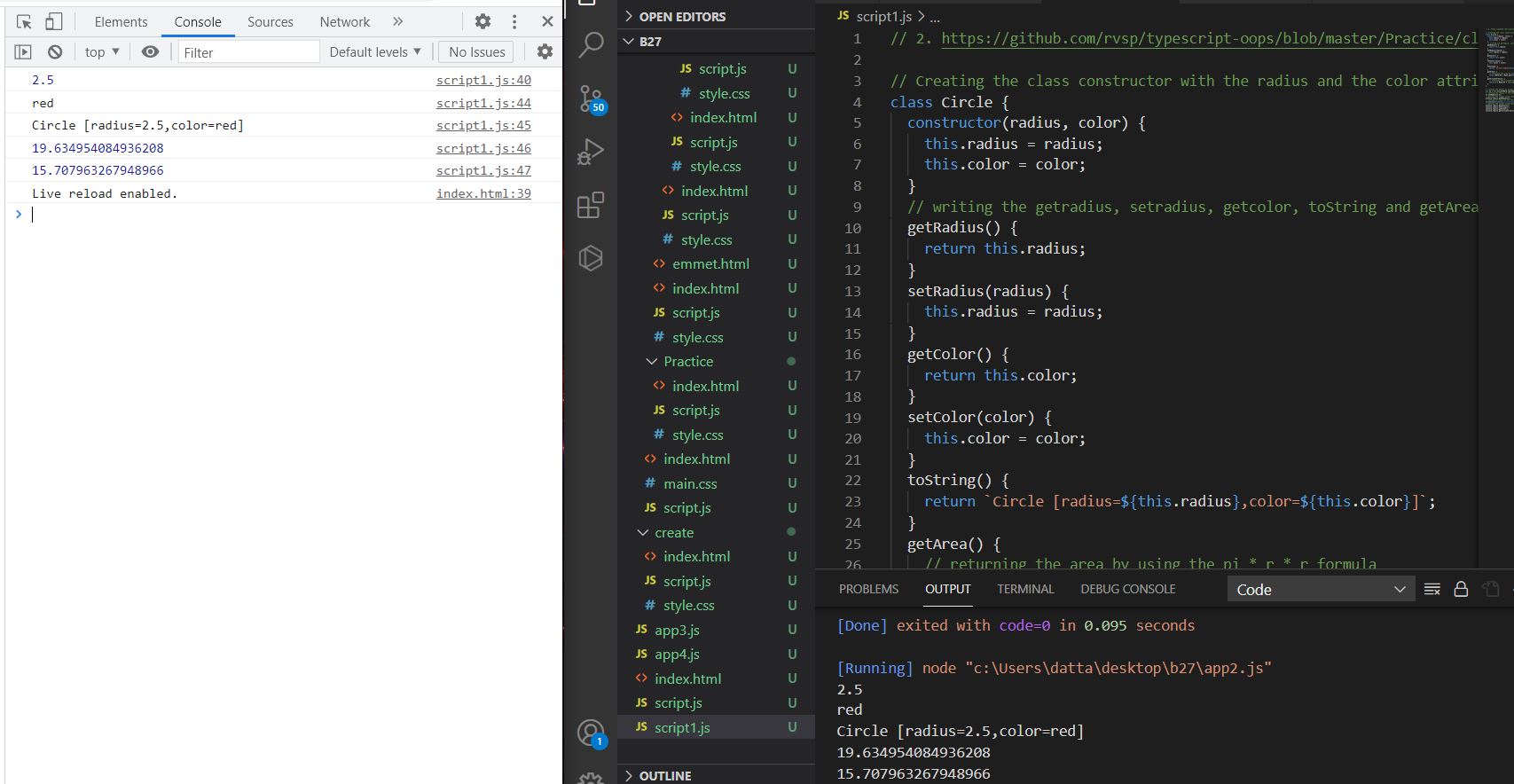
// getting the output using console.log()

console.log(c1.getColor());

console.log(c1.toString());

console.log(c1.getArea());

console.log(c1.getCircumference());

****

**3. Write a “person” class to hold all the details.**

// 3. Write a “person” class to hold all the details.

// Wring the person class with all the details for the required one.

class Person {

constructor(name, technology, experience\_years, email, phone, ...languages) {

this.name = name;

this.technology = technology;

this.experience\_years = experience\_years;

this.email = email;

this.phone = phone;

this.languages = languages;

}

}

// creating the details of the person object withe

let details = new Person(

"Dattatreya Mamidipaka",

"Full Stack Developer",

2,

"dattatreyamamidipaka@gmail.com",

9398342210,

"python",

"javascript",

"reactJS",

"nodeJS",

"mongoDB",

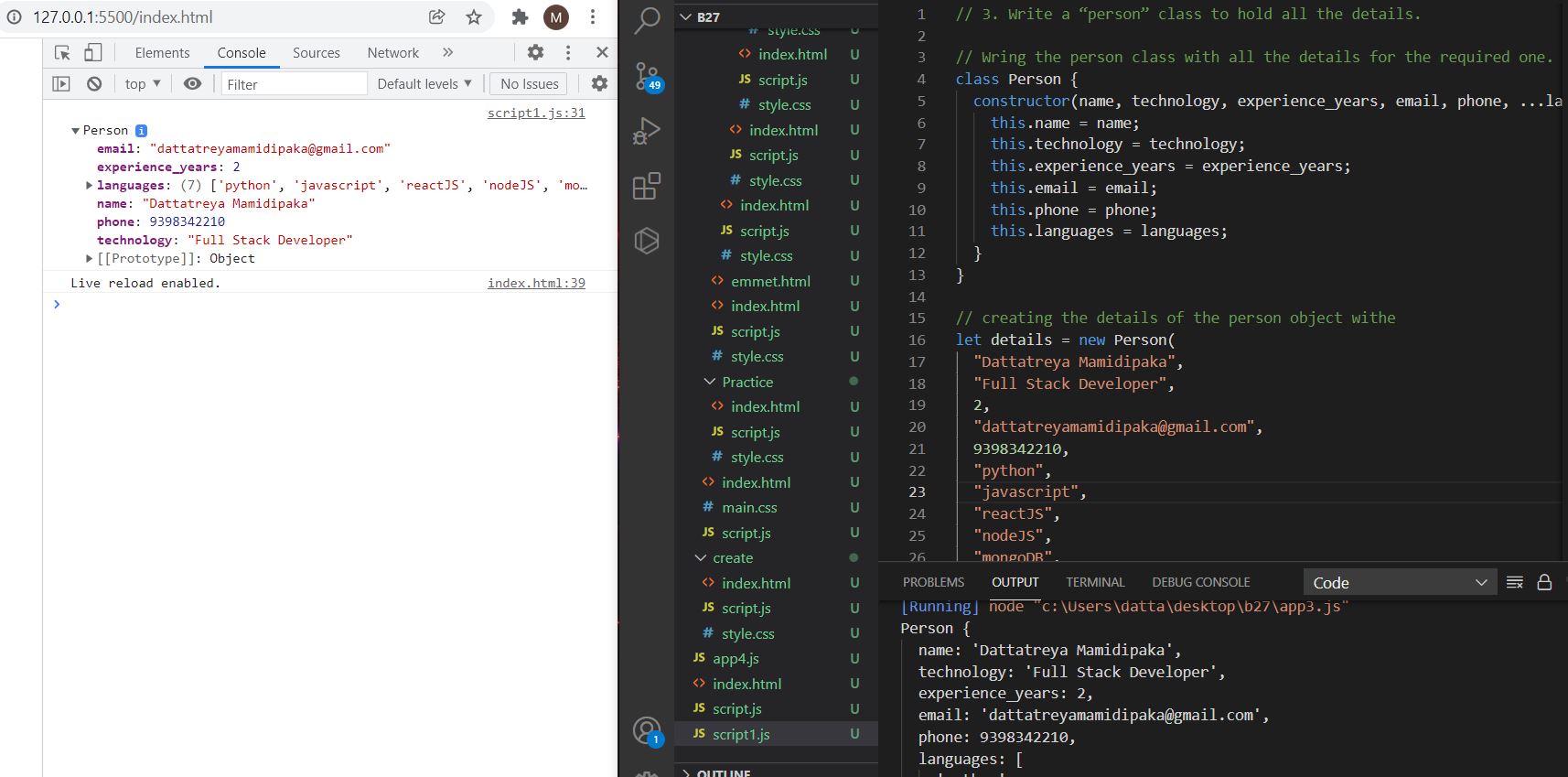
"HTML5",

"CSS3"

);

// printing the output of the class.

console.log(details);

****

**4. write a class to calculate uber price.**

// 4. write a class to calculate uber price.

// we are creating the constructor class and defining attributes.

class Uber {

constructor(car\_type, km, ac = false) {

this.car\_type = car\_type;

this.km = km;

this.ac = ac;

this.service\_charge = 12;

this.booking\_fee = 10;

this.tax = this.service\_charge + this.booking\_fee;

}

// get the price per km based on car type

getprice() {

// writing if condition if it is true add 2rs per each kms.

if (this.ac !== false) this.ac = 2;

else this.ac = 0;

if (this.car\_type == "primeSUV") return (10 + this.ac) \* this.km + this.tax;

else if (this.car\_type == "mini")

return (8.5 + this.ac) \* this.km + this.tax;

return (8 + this.ac) \* this.km + this.tax;

}

}

// taken the one constructor objects which take the values of the each the argument and printing total price.

let total\_price = new Uber("mini", 10, true);

console.log(total\_price.getprice());

let total\_price1 = new Uber("primeSUV", 20);

console.log(total\_price1.getprice());

