*const* express= require("express");

//databse

*const* database =require("./database");//using ./ means that databse is inthe same file

//initalize express

*const* booky =express();

/\*

 route  /

 description- get all teh bboks

 access - public

 parameter -none

 method - get

\*/

booky.get("./", (*req*,*res*) *=>* {  //here route is the root route

    return *res*.json({books: databse.books});

});

/\*

 route  /is

 description- get specific book on isbn

 access - public

 parameter -isbn

 method - get

\*/

booky.get("./is/:isbn", (*req*,*res*) *=>* {

*const* getspecificbook=database.books.filter(

        (*book*) *=>* *book*.ISBN ===*req*.params.isbn

    );

    if (getspecificbook.length===0){

        return *res*.json({error:`no bbok is found for isbn of $(req.params.isbn)`})

    }

    return *res*.json({books: getspecificbook});

});

/\*

 route  /c

 description- get specific book on category

 access - public

 parameter -category

 method - get

\*/

booky.get("/c/:category", (*req*,*res*) *=>* {

*const* getspecificbook=database.books.filter(

        (*book*) *=>* *book*.category.includes(*req*.params.category)

   );

    if (getspecificbook.length===0){

        return *res*.json({error:`no bbok is found for category of $(req.params.category)`})

    }

    return *res*.json({books: getspecificbook});

});

/\*

 route  /author

 description- get authors

 access - public

 parameter -none

 method - get

\*/

booky.get("/author", (*req*,*res*) *=>* {

    return *res*.json({authors: database.author});

});

/\*

 route  /author/book

 description- get authors based on books

 access - public

 parameter -isbn

 method - get

\*/

booky.get("/author/book/isbn", (*req*,*res*) *=>* {

*const* getspecificauthor = database.author.filter(

        (*author*) *=>* *author*.books.includes(*req*.params.isbn)

    );

    if (getspecificauthor.length===0){

        return *res*.json ({

            error: `no author found for the book of ${*req*.params.isbn}`

        });

    }

    return *res*.json({authors: getspecificauthor});

});

/\*

 route  /publication

 description- get all publications

 access - public

 parameter -none

 method - get

\*/

booky.get("/publication", (*req*,*res*) *=>* {

    return *res*.json({publication: database.publication});

});

booky.listen(3000,() *=>* {

    console.log("server is up and running");

});

*const* books =[

    {

    ISBN:"12345book",

    title:"tesla",

    pubdate:"2021-08-05",

    language:"en",

    numpage:"250",

    author:[1,2],

    publication:[1],

    category:["tech","space","education"],

    }

]

*const* author =[

    {

        id:1,

        name:"aradhana",

        books:["12345book",secretbook]

    },

    {

        id:2,

        name:"elon musk",

        books:["12345book"]

    }

]

*const* publication =[

    {

        id:1,

        name:"writex",

        books:["12345book"]

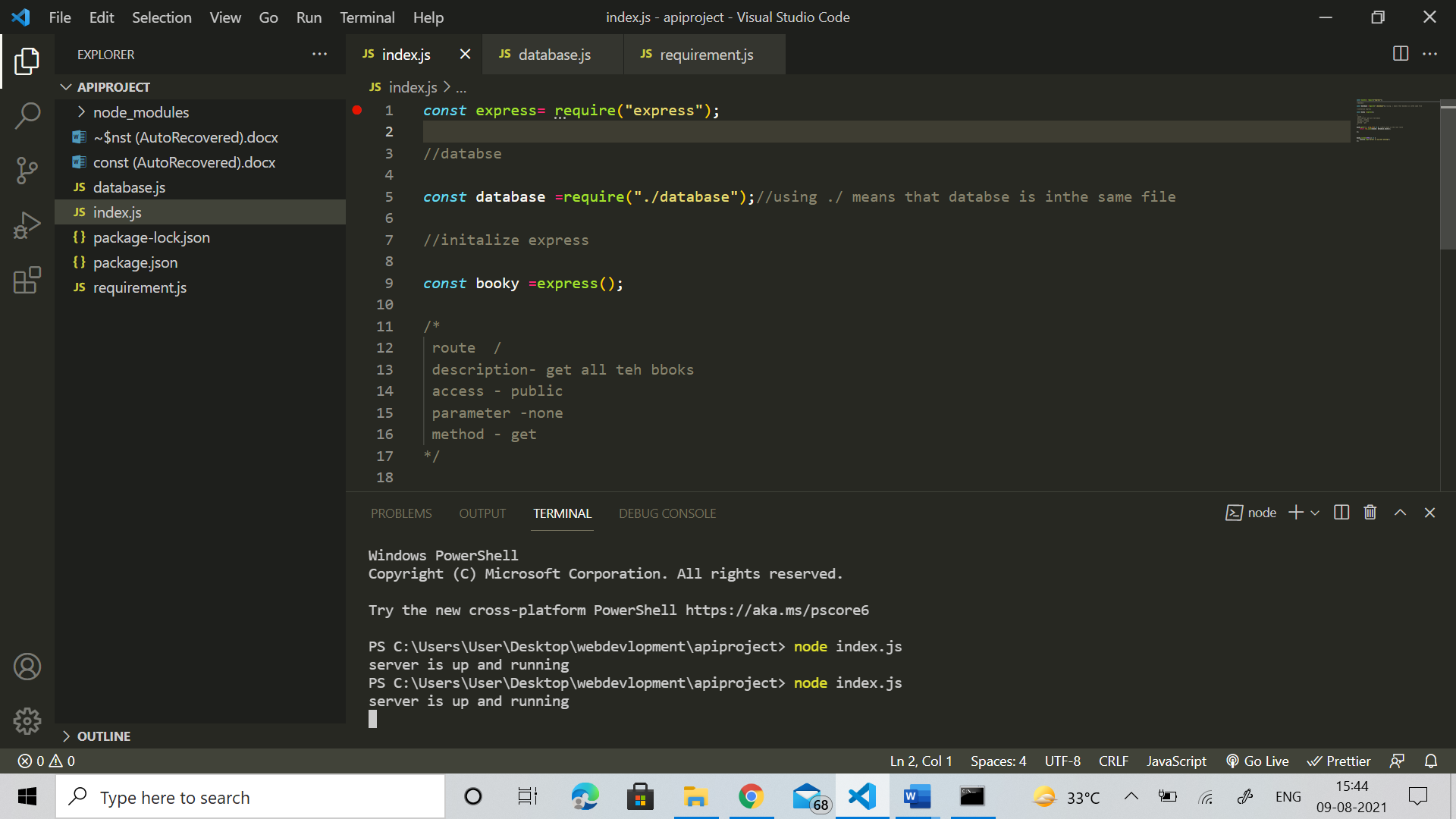
    }

]

*module* .exports = {books,author,publication}// this says this module exports the contenrt between the parenthesis

cd webevlopment/apiproject

node index.js



**const** *books* =[

    {

*ISBN*:"12345book",

*title*:"tesla",

*pubdate*:"2021-08-05",

*language*:"en",

*numpage*:"250",

*author*:[1,2],

*publication*:[1],

*category*:["tech","space","education"],

    }

]

**const** *author* =[

    {

*id*:1,

*name*:"aradhana",

*books*:["12345book",*secretbook*]

    },

    {

*id*:2,

*name*:"elon musk",

*books*:["12345book"]

    }

]

**const** *publication* =[

    {

*id*:1,

*name*:"writex",

*books*:["12345book"]

    }

]

module .*exports* = {*books*,*author*,*publication*}*//this says this module expots the contenrt between the parenthesis*

*//requirement for our project*

*// books*

*//isbn,title,pub data,languaage,num page,array of authors [], category []*

*//author id ,name, books*

*//publications*

*//ID,name,books[]*