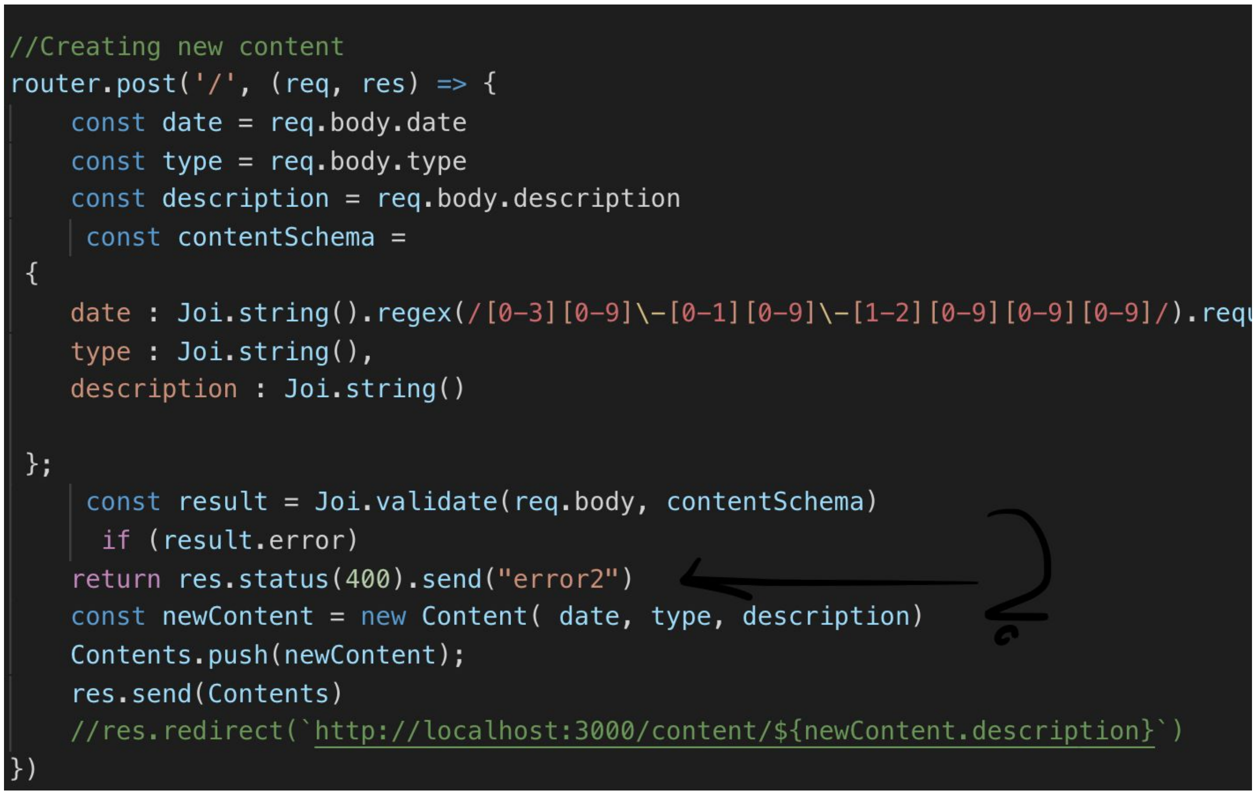
Sprint Backlog	Completeness	Estimation Report	Best practises (Naming conventions, indentations, code structure, etc..)	Integration on master (Individual)
1- The backlog itself (0 or 1) 1- The fair allocation among team members (0 or 1) 2/2	0 - Less than 30% of the sprint backlog 1 - 30-80% 2 - 80-100% 2/3	1 - Team velocity (0 none, 0.5 fake, 1 realistic) 1 - Sprint burndown chart (0 none, 0.5 fake, 1 realistic) 2/2	0 - Spaghetti code 1- Adhered to the main structure and naming conventions 2- Clean code 0/1	0 - Code not integrated 2 - Code integrated 2/2

Comments:

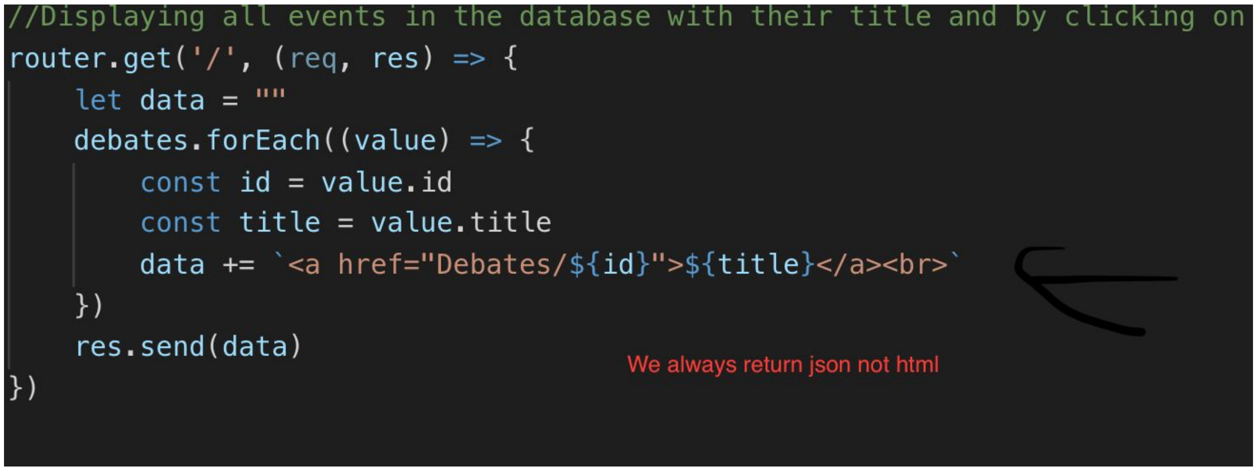
1. Debates API is not following the naming conventions
2. Naming of files in schemas folder is not uniform
3. Indentations are not uniform
4. FAQs, and users no validations, not following the rest convention
5. User has two puts?
6. Where are the mates user stories?

7. You all got the integration grade



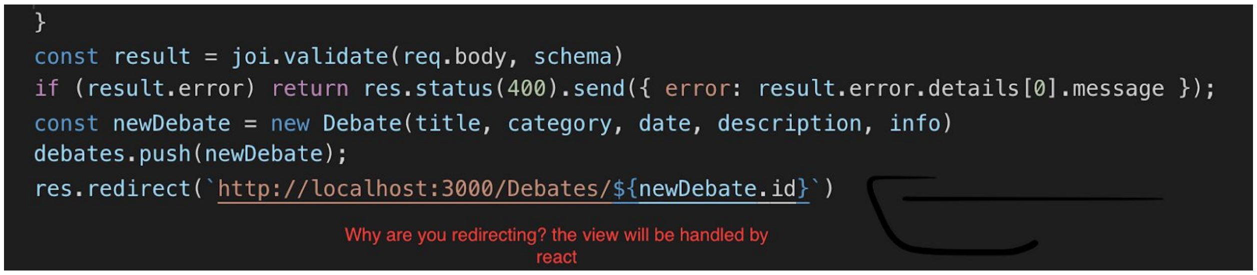
```
//Creating new content
router.post('/', (req, res) => {
    const date = req.body.date
    const type = req.body.type
    const description = req.body.description
     const contentSchema =
 {
    date : Joi.string().regex(/[0-3][0-9]\-[0-1][0-9]\-[1-2][0-9][0-9][0-9]/).requ
    type : Joi.string(),
    description : Joi.string()

 };
     const result = Joi.validate(req.body, contentSchema)
      if (result.error)
    return res.status(400).send("error2")
    const newContent = new Content( date, type, description)
    Contents.push(newContent);
    res.send(Contents)
    //res.redirect(`http://localhost:3000/content/${newContent.description}`)
})
```

?



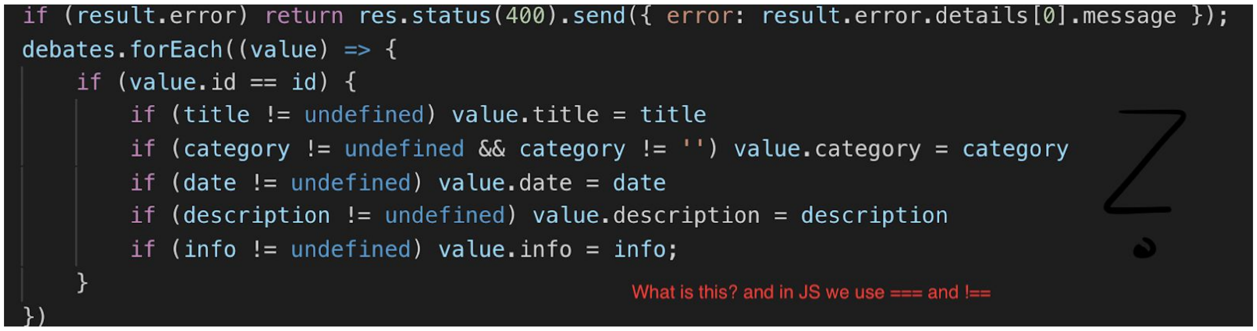
```
//Displaying all events in the database with their title and by clicking on
router.get('/', (req, res) => {
    let data = ""
    debates.forEach((value) => {
        const id = value.id
        const title = value.title
        data += `<a href="Debates/${id}">${title}</a><br>`
    })
    res.send(data)
})
```

We always return json not html



```
  }
  const result = joi.validate(req.body, schema)
  if (result.error) return res.status(400).send({ error: result.error.details[0].message });
  const newDebate = new Debate(title, category, date, description, info)
  debates.push(newDebate);
  res.redirect(`http://localhost:3000/Debates/${newDebate.id}`)
```

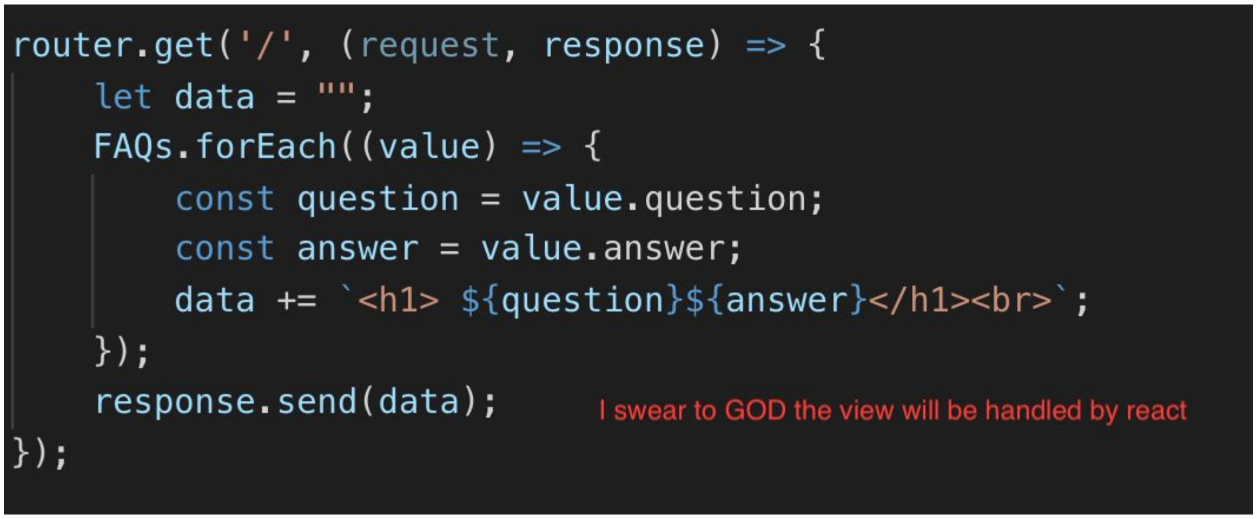
Why are you redirecting? the view will be handled by react



```
if (result.error) return res.status(400).send({ error: result.error.details[0].message });
debates.forEach((value) => {
    if (value.id == id) {
        if (title != undefined) value.title = title
        if (category != undefined && category != '') value.category = category
        if (date != undefined) value.date = date
        if (description != undefined) value.description = description
        if (info != undefined) value.info = info;
    }
})
```

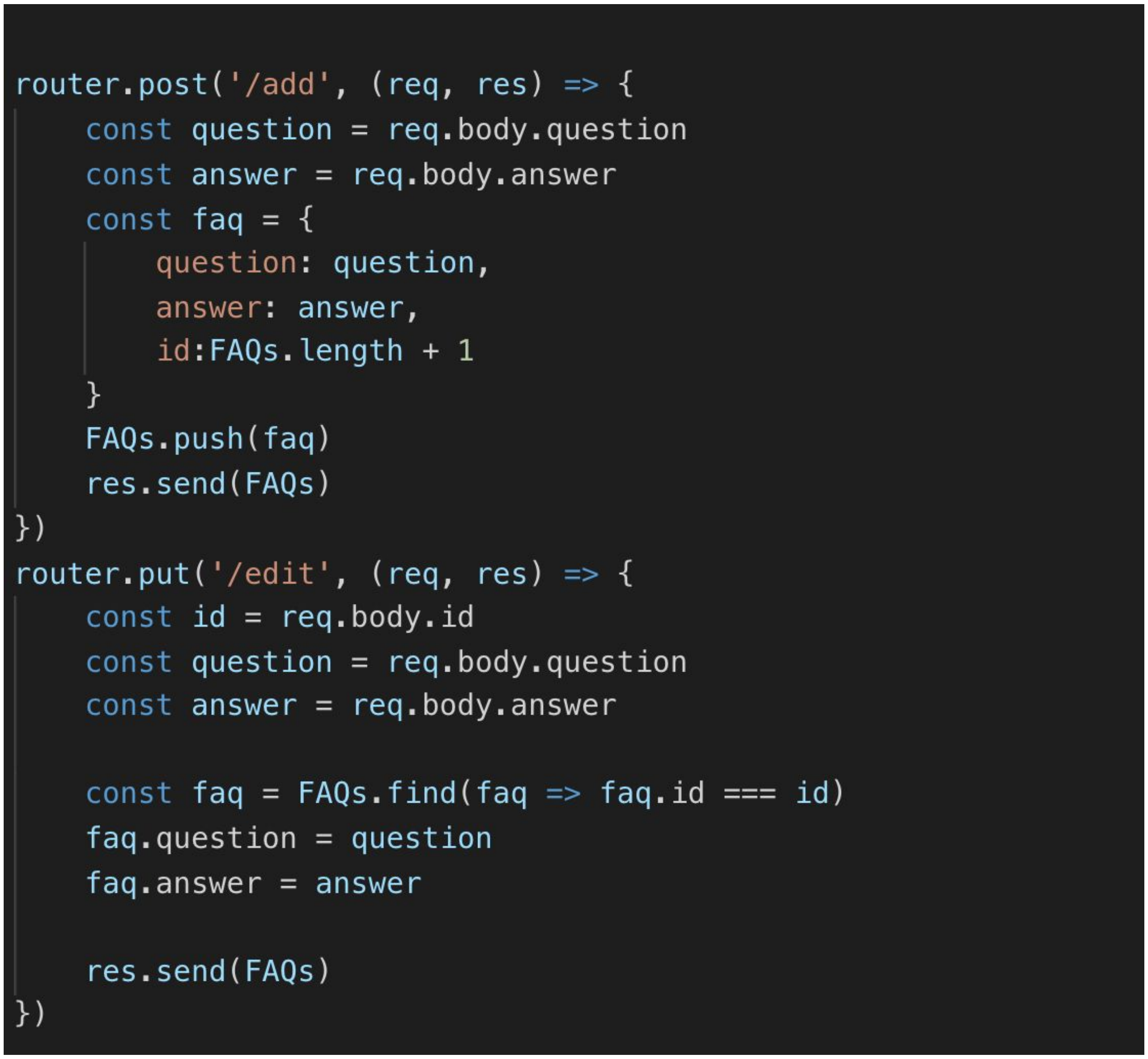
¿

What is this? and in JS we use === and !==

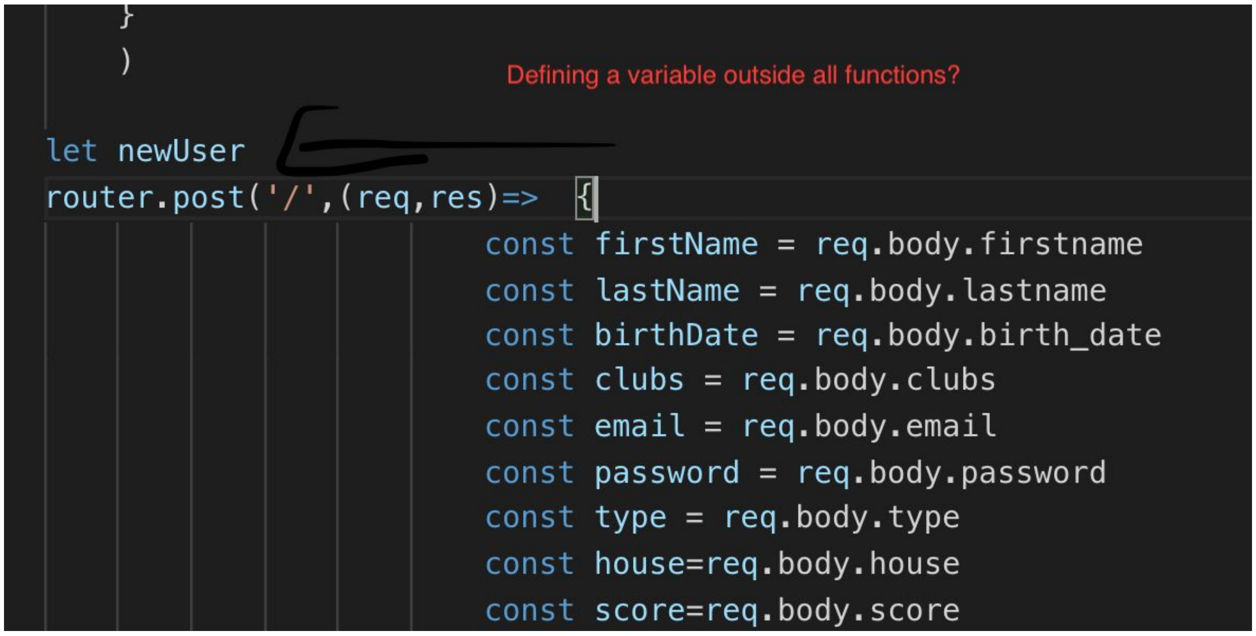


```
router.get('/', (request, response) => {
    let data = "";
    FAQs.forEach((value) => {
        const question = value.question;
        const answer = value.answer;
        data += `<h1> ${question}${answer}</h1><br>`;
    });
    response.send(data);
});
```

I swear to GOD the view will be handled by react

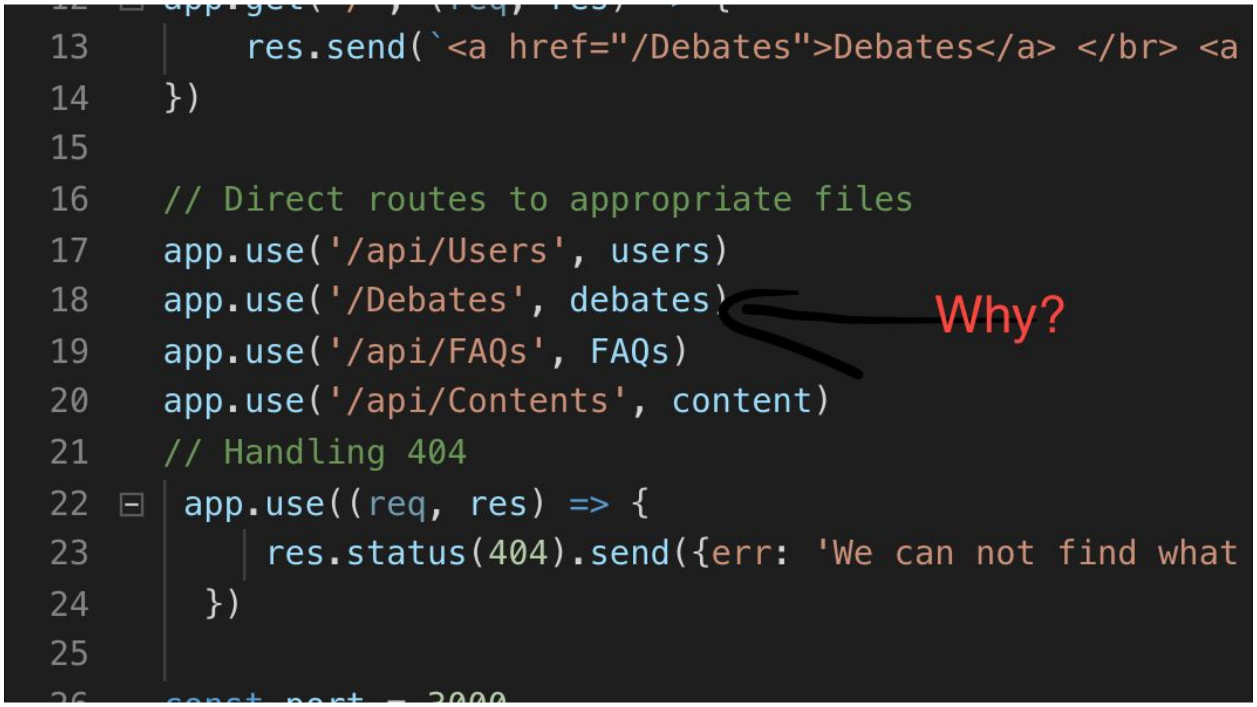


```javascript
router.post('/add', (req, res) => {
    const question = req.body.question
    const answer = req.body.answer
    const faq = {
        question: question,
        answer: answer,
        id:FAQs.length + 1
    }
    FAQs.push(faq)
    res.send(FAQs)
})
router.put('/edit', (req, res) => {
    const id = req.body.id
    const question = req.body.question
    const answer = req.body.answer

    const faq = FAQs.find(faq => faq.id === id)
    faq.question = question
    faq.answer = answer

    res.send(FAQs)
})
```

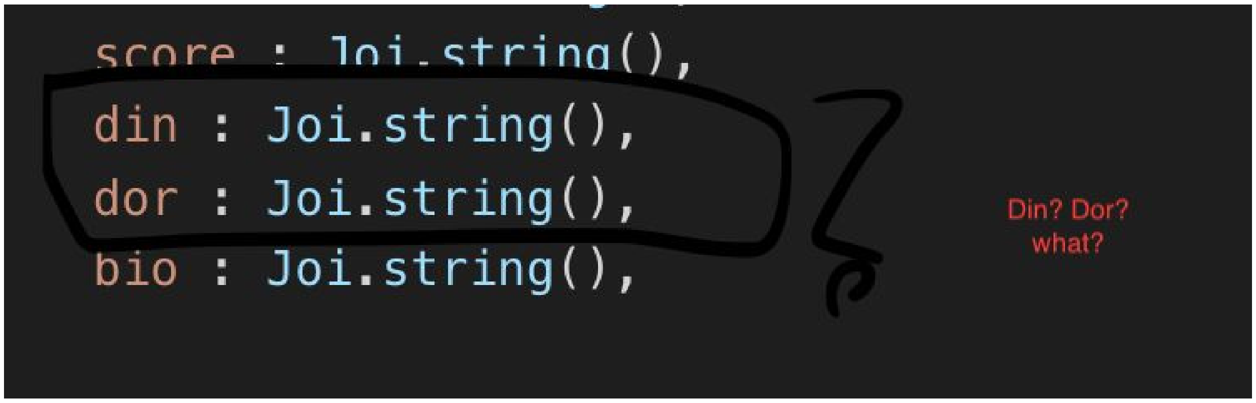


Defining a variable outside all functions?

```
    }
    )

let newUser
router.post('/',(req,res)=>  {
                        const firstName = req.body.firstname
                        const lastName = req.body.lastname
                        const birthDate = req.body.birth_date
                        const clubs = req.body.clubs
                        const email = req.body.email
                        const password = req.body.password
                        const type = req.body.type
                        const house=req.body.house
                        const score=req.body.score
```



```
    res.send(`<a href="/Debates">Debates</a> </br> <a
})

// Direct routes to appropriate files
app.use('/api/Users', users)
app.use('/Debates', debates)
app.use('/api/FAQs', FAQs)
app.use('/api/Contents', content)
// Handling 404
 app.use((req, res) => {
     res.status(404).send({err: 'We can not find what
  })

```



```
score : Joi.string(),
din : Joi.string(),
dor : Joi.string(),
bio : Joi.string(),
```