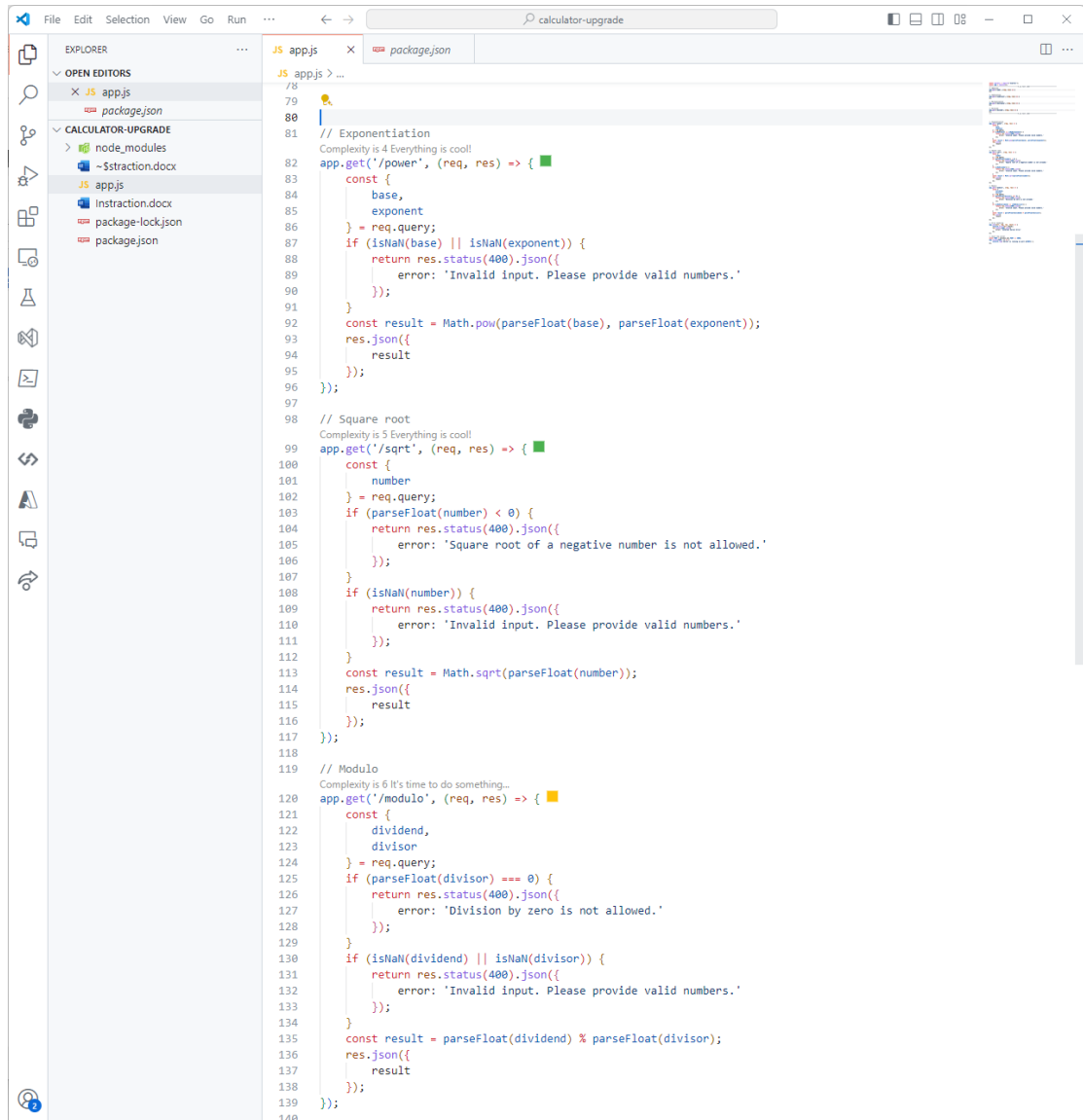


GitHub link: <https://github.com/DennisPYu/sit323-2024-t1-prac4c>

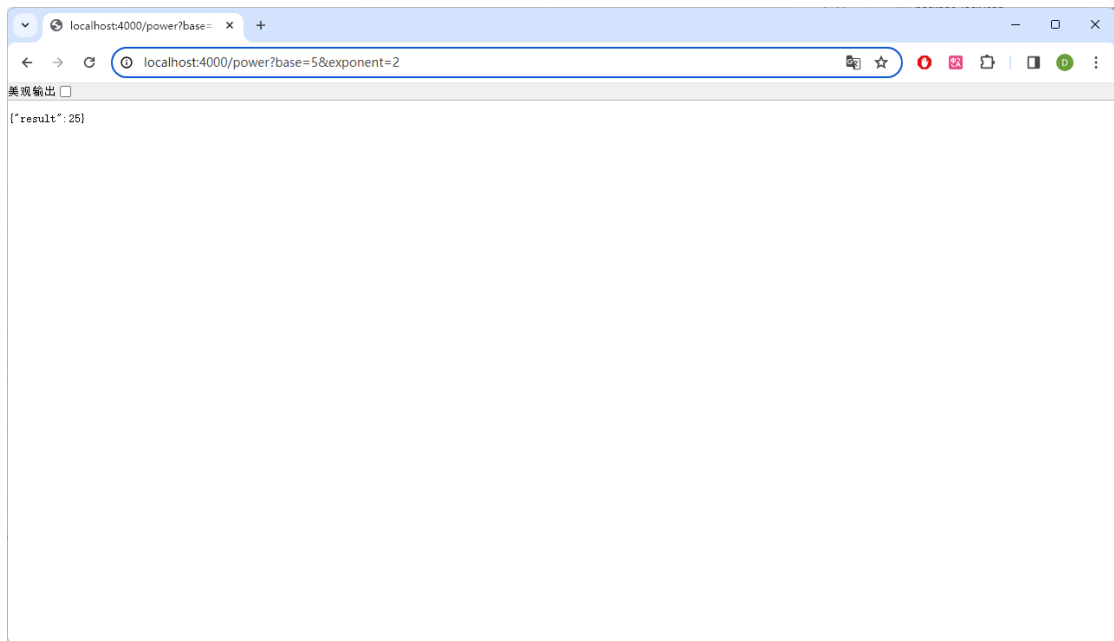
Note: this task is the extend version of task 4.1p. It contains some of the code that I used in task 4.1p.

- Code

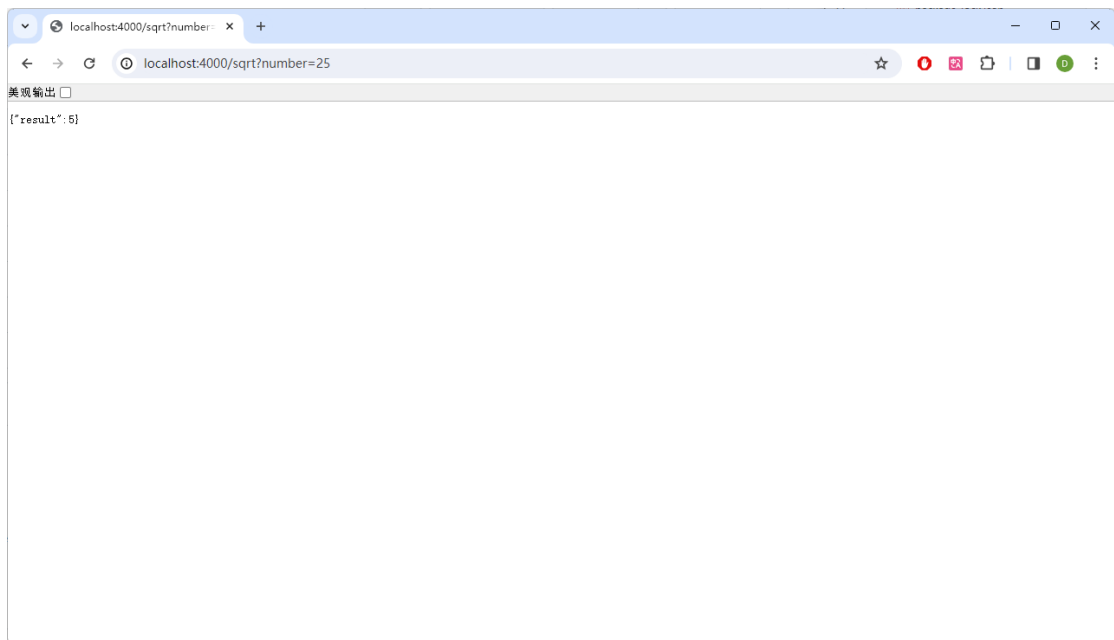


```
78
79
80
81 // Exponentiation
82 // Complexity is 4 Everything is cool!
83 app.get('/power', (req, res) => {
84   const {
85     base,
86     exponent
87   } = req.query;
88   if (isNaN(base) || isNaN(exponent)) {
89     return res.status(400).json({
90       error: 'Invalid input. Please provide valid numbers.'
91     });
92   }
93   const result = Math.pow(parseFloat(base), parseFloat(exponent));
94   res.json({
95     result
96   });
97 });
98
99 // Square root
100 // Complexity is 5 Everything is cool!
101 app.get('/sqrt', (req, res) => {
102   const {
103     number
104   } = req.query;
105   if (parseFloat(number) < 0) {
106     return res.status(400).json({
107       error: 'Square root of a negative number is not allowed.'
108     });
109   }
110   if (isNaN(number)) {
111     return res.status(400).json({
112       error: 'Invalid input. Please provide valid numbers.'
113     });
114   }
115   const result = Math.sqrt(parseFloat(number));
116   res.json({
117     result
118   });
119 });
120
121 // Modulo
122 // Complexity is 6 It's time to do something...
123 app.get('/modulo', (req, res) => {
124   const {
125     dividend,
126     divisor
127   } = req.query;
128   if (parseFloat(divisor) === 0) {
129     return res.status(400).json({
130       error: 'Division by zero is not allowed.'
131     });
132   }
133   if (isNaN(dividend) || isNaN(divisor)) {
134     return res.status(400).json({
135       error: 'Invalid input. Please provide valid numbers.'
136     });
137   }
138   const result = parseFloat(dividend) % parseFloat(divisor);
139   res.json({
140     result
141   });
142 });
143
144
```

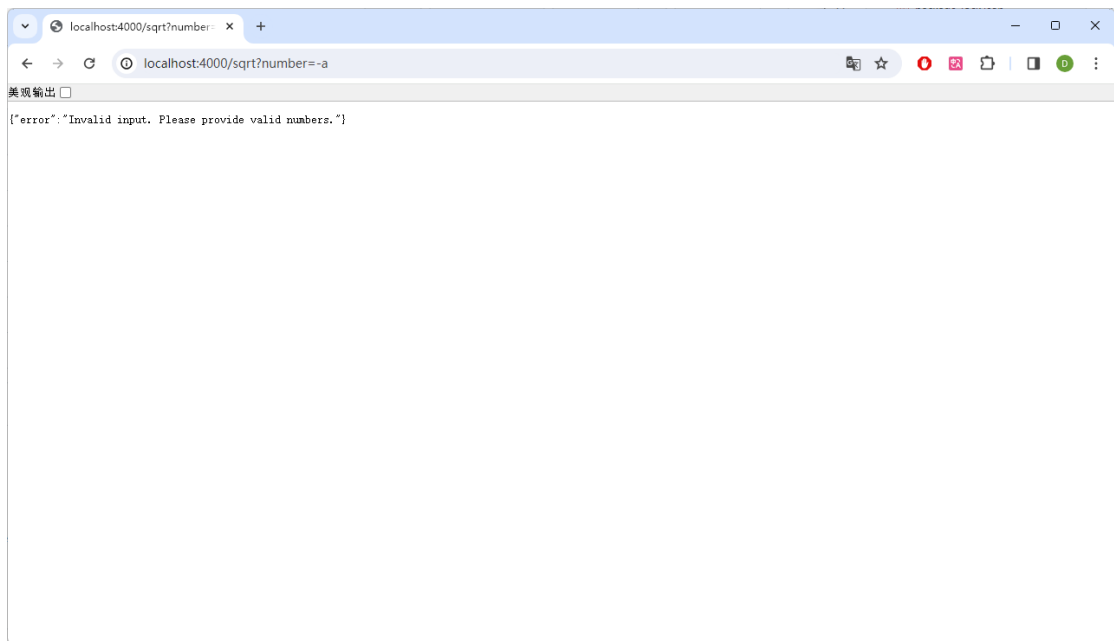
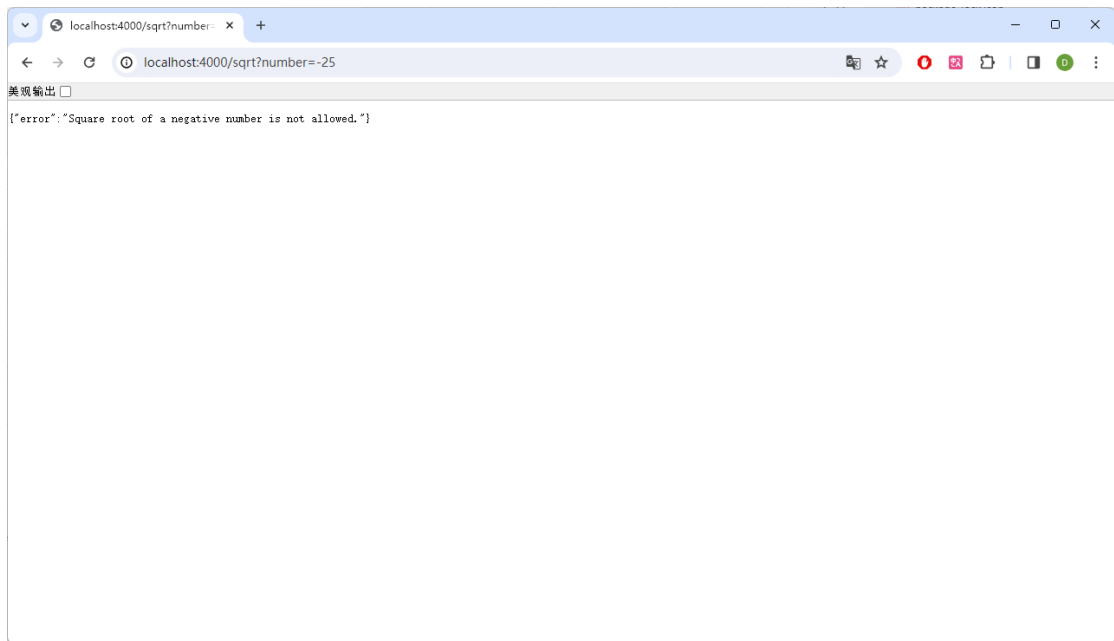
- Exponentiation



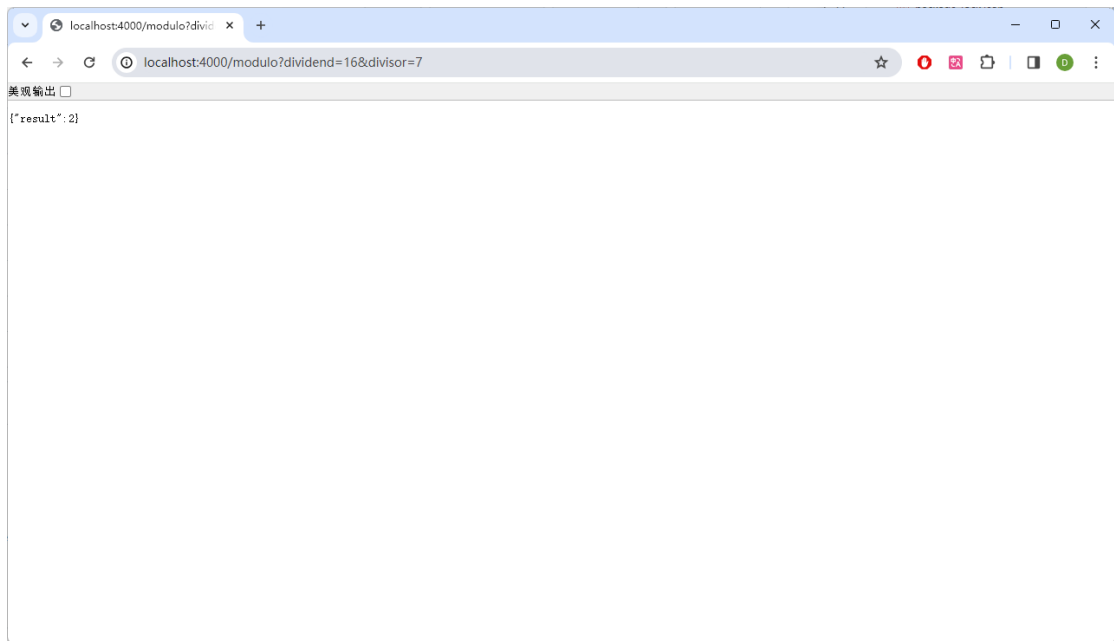
- Square root



- Square root      error      handling



- Modulo



- Modulo error handling

