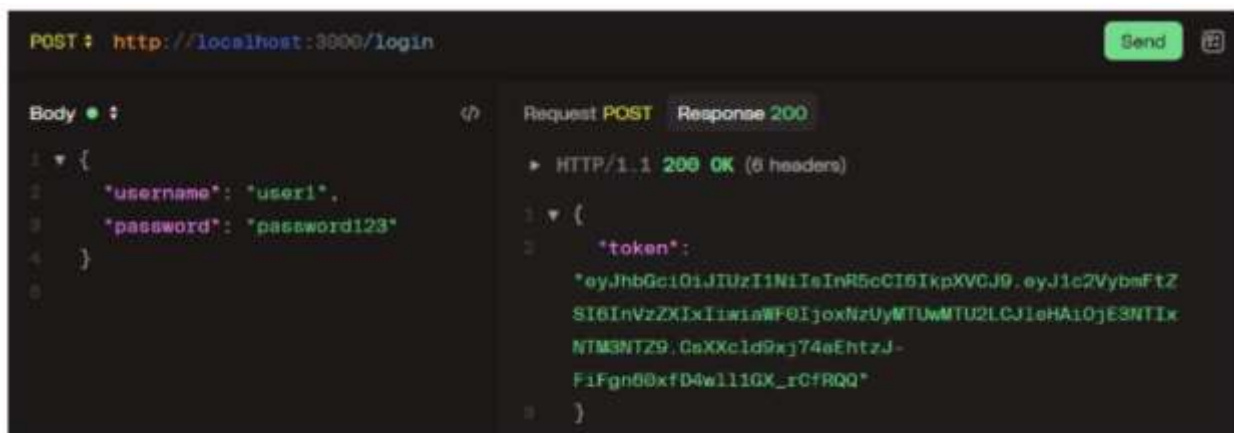| NAME | Vikash kr jha |
|------|---------------|
| UID | 23BCS11391 |
| CLASS | 622-A |

➢ NodeJs PRACTISE 6.2

• CODE

```javascript
const express = require('express')
const jwt = require('jsonwebtoken')
const app = express()
app.use(express.json())

const SECRET_KEY = 'mysecretkey'
let balance = 1000

app.post('/login', (req, res) => {
const { username, password } = req.body
if (username === 'alpha' && password === 'alpha2025') {
const token = jwt.sign({ username }, SECRET_KEY, { expiresIn: '1h' })
res.json({ token })
} else {
res.status(401).json({ message: 'Invalid credentials' })
}
})

function authenticateToken(req, res, next) {
const authHeader = req.headers['authorization']
const token = authHeader && authHeader.split(' ')[1]
if (!token) return res.status(403).json({ message: 'Token required' })
jwt.verify(token, SECRET_KEY, (err, user) => {
```

```javascript
if (err) return res.status(403).json({ message: 'Invalid or expired token'
})
req.user = user
next()
})
}

app.get('/balance', authenticateToken, (req, res) => {
res.json({ balance })
})

app.post('/deposit', authenticateToken, (req, res) => {
const { amount } = req.body
if (amount > 0) {
balance += amount
res.json({ balance })
} else {
res.status(400).json({ message: 'Invalid deposit amount' })
}
})

app.post('/withdraw', authenticateToken, (req, res) => {
const { amount } = req.body
if (amount > balance) return res.status(400).json({ message: 'Insufficient
balance' })
if (amount <= 0) return res.status(400).json({ message: 'Invalid
withdrawal amount' })
balance -= amount
res.json({ balance })
})

app.listen(3000, () => console.log('Server running on port 3000'))
```

OUTPUT:

POST: http://localhost:3000/login                                    Send

Body • :                          </>        Request POST  Response 200

1 ▾ {                                        ▸ HTTP/1.1 200 OK (6 headers)
2     "username": "user1",
3     "password": "password123"             1 ▾ {
4   }                                        2     "token":
                                                "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZ
                                                SI6InVzZXIxIiwiaWF0IjoxNzUyMTUwMTU2LCJIeHAiOjE3NTIx
                                                NTM3NTZ9.CsXXc1d9xj74aEhtzJ-
                                                FiFgn00xfD4w1l1GX_rCfRQQ"
                                             3   }

## GET http://localhost:3000/balance

**Body**

Request GET | Response 403

▸ HTTP/1.1 **403 Forbidden** (6 headers)

```
1 ▾ {
2     "message": "Invalid or expired token"
3   }
```

## GET http://localhost:3000/balance

**Auth**

eyJhbGciOiJIUzI1NiIsInR5cCI6Ikp
XVCJ9.eyJ1c2VybmFtZSI6InVzZXIxI
iwiaWF0IjoxNzUyMTUwMTU2LCJleHAi
OjE3NTIxNTM3NTZ9.CsXXcld9xj74aE
htzJ-FiFgn60xfD4wll1GX_rCfRQQ

Request GET | Response 200

▸ HTTP/1.1 **200 OK** (6 headers)

```
1 ▾ {
2     "balance": 1000
3   }
```

## POST http://localhost:3000/deposit

**Body**

```
1 ▾ {
2     "amount": 250
3   }
4
```

Request POST | Response 200

▸ HTTP/1.1 **200 OK** (6 headers)

```
1 ▾ {
2     "message": "Deposited $250",
3     "newBalance": 1250
4   }
```

## POST http://localhost:3000/withdraw

**Body**

```
1 ▾ {
2     "amount": 150
3   }
4
```

Request POST | Response 200

▸ HTTP/1.1 **200 OK** (6 headers)

```
1 ▾ {
2     "message": "Withdrew $150",
3     "newBalance": 1100
4   }
```