



Software Engineering
IT314: Lab8
Group 22

We are going to test 'login' module of our backend using the frameworks like mocha and chai.

Login utility function which we are going to test:

```
/**
 * @description Login Utility Function
 */
exports.findPerson = async (req, res) => {
  const email = req.body.email;
  const role = req.body.role;
  const password = req.body.password;

  if (role == "Student") {
    await student.find({ email: email })
      .then((data) => {
        if (!data) {
          res
            .status(404)
            .send({ message: `Not found user with email: ${email} ` });
        } else {
          if (!bcrypt.compareSync(password, data[0].password)) {
            res
              .status(500)
              .send({ message: `Password Invalid` });
          }
          return;
        }
        // create token
        const token = generateToken(data[0]._id, email, role);
        console.log(token);
        res.cookie("jwt", token, { maxAge: cookie_expires_in, httpOnly: true });
        res.send(data);
        // res.render('studentProfile', { student: data[0] });
      })
  }
}
```

```
    })
    .catch((err) => {
      res
        .status(500)
        .send({ message: `Error retrieving user with email ${email}` });
    });
  }
  else if (role == "Company") {
    await company.find({ email: email })
      .then((data) => {
        if (!data) {
          // Make new webpage for all not found errors
          res
            .status(404)
            .send({ message: `Not found Company with email: ${email}` });
        } else {
          if (!bcrypt.compareSync(password, data[0].password)) {
            res
              .status(500)
              .send({ message: `Password Invalid` });
            return;
          }
          // create token
          const token = generateToken(data[0]._id, email, role);
          console.log(token);
          res.cookie("jwt", token, { maxAge: cookie_expires_in, httpOnly: true });
          res.send(data);
          // res.render('companyProfile', { company: data[0] });
        }
      })
    .catch((err) => {
      res
        .status(500)
        .send({ message: `Error retrieving user with email ${email}` });
    });
  }
  else {
    await admin.find({ email: email })
      .then(async (data) => {
        {
          if (!data) {
            // Make new webpage for all not found errors
            res
              .status(404)
              .send({ message: `Not found admin with email: ${email}` });
          } else {
            // initialize cookie with role student and email
            if (data[0].role == 1) {
              if (role == "Placement Manager") {
                if (!bcrypt.compareSync(password, data[0].password)) {
                  res
                    .status(500)
                    .send({ message: `Password Invalid` });
                  return;
                }
              }
            }
          }
        }
      })
    .catch((err) => {
      res
        .status(500)
        .send({ message: `Error retrieving user with email ${email}` });
    });
  }
}
```

```
        // create token
        const token = generateToken(data[0]._id, email, role);
        console.log(token);
        res.cookie("jwt", token, { maxAge: cookie_expires_in, httpOnly:
true });

        // res.render('superAdminProfile', { admin: data[0] });
        res.send(data);
    }
    else {
        res
            .status(500)
            .send({ message: `Error retrieving user with role ${role}`
});
    }
}
else {
    if (role === "Admin") {
        if (!bcrypt.compareSync(password, data[0].password)) {
            res
                .status(500)
                .send({ message: `Password Invalid` });
            return;
        }
        // create token
        const token = generateToken(data[0]._id, email, role);
        console.log(token);
        res.cookie("jwt", token, { maxAge: cookie_expires_in, httpOnly:
true });

        // res.render('adminProfile', { admin: data[0] });
        res.send(data);
    }
    else {
        res
            .status(500)
            .send({ message: `Error retrieving user with role ${role}`
});
    }
}
}
})
.catch((err) => {
    res
        .status(500)
        .send({ message: `Error retrieving user with email ${email}` });
});
}
};
```

Testcases:

1. Role Student: With All valid Credentials
2. Role Student: Student email does not contain in our database
3. Role Student: With null email address
4. Role Student: With Wrong Password
5. Role Company: With All valid Credentials
6. Role Company: Company email does not contain in our database
7. Role Company: With null email address
8. Role Company: With Wrong Password
9. Role Admin: With All valid Credentials
10. Role Admin: Admin email does not contain in our database
11. Role Admin: With null email address
12. Role Admin: With Wrong Password
13. Role Placement Manager: With All valid Credentials
14. Role Placement Manager: With Wrong Password
15. Logging in with Student Email but Role is Admin
16. Logging in with Company Email but Role is Student
17. Logging in with Admin Email but Role is Company
18. Logging in with Placement Manager Email but Role is Student

Testing File Content:

```
let chai = require("chai");
let chaiHttp = require("chai-http");

// Assertion Style
chai.should();
chai.use(chaiHttp);

describe('Testing route /login', () =>
{
  const host = `http://localhost:3000`;
  const path = "/login";

  // Role Student
  it("Role Student: With All valid Credentials", (done) =>
  {
    chai
      .request(host)
      .get(path)
      .set('content-type', 'application/x-www-form-urlencoded')
      .send({ email: '202001226@daict.ac.in', password: 'jatin1510', role: 'Student'
    })
  })
}
```

```
.end(function (error, response, body)
{
    response.should.have.status(200);
    done();
});

});
it("Role Student: Student email does not contain in our database", (done) =>
{
    chai
        .request(host)
        .get(path)
        .set('content-type', 'application/x-www-form-urlencoded')
        .send({ email: '202001258@daict.ac.in', password: 'jatin1510', role: 'Student'
    })

    .end(function (error, response, body)
    {
        response.should.have.status(500);
        done();
    });

});

it("Role Student: With null email address", (done) =>
{
    chai
        .request(host)
        .get(path)
        .set('content-type', 'application/x-www-form-urlencoded')
        .send({ email: '', password: 'jatin1510', role: 'Student' })
        .end(function (error, response, body)
        {
            response.should.have.status(500);
            done();
        });

});

it("Role Student: With Wrong Password", (done) =>
{
    chai
        .request(host)
        .get(path)
        .set('content-type', 'application/x-www-form-urlencoded')
        .send({ email: '202001226@daict.ac.in', password: 'jatin15102002', role:
'Student' })
        .end(function (error, response, body)
        {
            response.should.have.status(500);
            done();
        });

});

// Role Student
it("Role Company: With All valid Credentials", (done) =>
{
    chai
```

```
.request(host)
.get(path)
.set('content-type', 'application/x-www-form-urlencoded')
.send({ email: 'google@gmail.com', password: 'google', role: 'Company' })
.end(function (error, response, body)
{
    response.should.have.status(200);
    done();
});

});

it("Role Company: Company email does not contain in our database", (done) =>
{
    chai
    .request(host)
    .get(path)
    .set('content-type', 'application/x-www-form-urlencoded')
    .send({ email: 'mocha@gmail.com', password: 'mochi', role: 'Company' })
    .end(function (error, response, body)
    {
        response.should.have.status(500);
        done();
    });

});

it("Role Company: With null email address", (done) =>
{
    chai
    .request(host)
    .get(path)
    .set('content-type', 'application/x-www-form-urlencoded')
    .send({ email: '', password: 'password', role: 'Company' })
    .end(function (error, response, body)
    {
        response.should.have.status(500);
        done();
    });

});

it("Role Company: With Wrong Password", (done) =>
{
    chai
    .request(host)
    .get(path)
    .set('content-type', 'application/x-www-form-urlencoded')
    .send({ email: 'google@gmail.com', password: 'googlers', role: 'Company' })
    .end(function (error, response, body)
    {
        response.should.have.status(500);
        done();
    });

});

// Role Admin
it("Role Admin: With All valid Credentials", (done) =>
```

```
{
  chai
    .request(host)
    .get(path)
    .set('content-type', 'application/x-www-form-urlencoded')
    .send({ email: '202003039@daiict.ac.in', password: 'iammukli', role: 'Admin' })
    .end(function (error, response, body)
    {
      response.should.have.status(200);
      done();
    });

});

it("Role Admin: Admin email does not contain in our database", (done) =>
{
  chai
    .request(host)
    .get(path)
    .set('content-type', 'application/x-www-form-urlencoded')
    .send({ email: '202001196@daiict.ac.com', password: 'viraj', role: 'Admin' })
    .end(function (error, response, body)
    {
      response.should.have.status(500);
      done();
    });

});

it("Role Admin: With null email address", (done) =>
{
  chai
    .request(host)
    .get(path)
    .set('content-type', 'application/x-www-form-urlencoded')
    .send({ email: '', password: 'password', role: 'Admin' })
    .end(function (error, response, body)
    {
      response.should.have.status(500);
      done();
    });

});

it("Role Admin: With Wrong Password", (done) =>
{
  chai
    .request(host)
    .get(path)
    .set('content-type', 'application/x-www-form-urlencoded')
    .send({ email: '202003039@daiict.ac.in', password: 'iamMukundLadani', role:
'Admin' })
    .end(function (error, response, body)
    {
      response.should.have.status(500);
      done();
    });

});
```

```
// Role Placement Manager
it("Role Placement Manager: With All valid Credentials", (done) =>
{
    chai
        .request(host)
        .get(path)
        .set('content-type', 'application/x-www-form-urlencoded')
        .send({ email: 'placement@daiict.ac.in', password: 'superadmin', role:
'Placement Manager' })
        .end(function (error, response, body)
        {
            response.should.have.status(200);
            done();
        });

});

it("Role Placement Manager: With Wrong Password", (done) =>
{
    chai
        .request(host)
        .get(path)
        .set('content-type', 'application/x-www-form-urlencoded')
        .send({ email: 'placement@daiict.ac.in', password: 'notsuperadmin', role:
'Placement Manager' })
        .end(function (error, response, body)
        {
            response.should.have.status(500);
            done();
        });

});

// Role changes for login
it("Logging in with Student Email but Role is Admin", (done) =>
{
    chai
        .request(host)
        .get(path)
        .set('content-type', 'application/x-www-form-urlencoded')
        .send({ email: '202001226@daiict.ac.in', password: 'jatin1510', role: 'Admin' })
        .end(function (error, response, body)
        {
            response.should.have.status(500);
            done();
        });

});

it("Logging in with Company Email but Role is Student", (done) =>
{
    chai
        .request(host)
        .get(path)
        .set('content-type', 'application/x-www-form-urlencoded')
        .send({ email: 'google@gmail.com', password: 'google', role: 'Student' })
        .end(function (error, response, body)
        {
            response.should.have.status(500);
            done();
        });

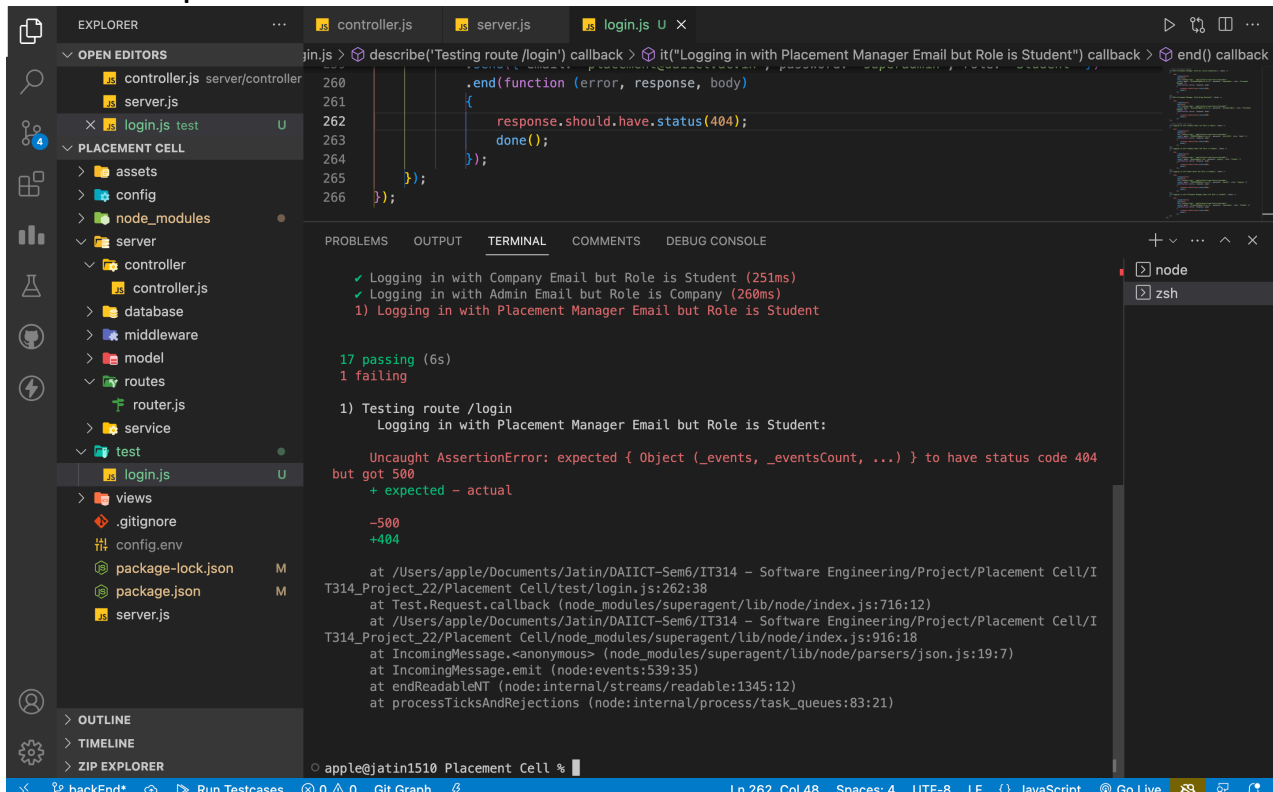
});
```



```
        response.should.have.status(500);
        done();
    });
});
it("Logging in with Admin Email but Role is Company", (done) =>
{
    chai
        .request(host)
        .get(path)
        .set('content-type', 'application/x-www-form-urlencoded')
        .send({ email: '202003039@daiict.ac.in', password: 'immukli', role: 'Company' })
        .end(function (error, response, body)
        {
            response.should.have.status(500);
            done();
        });
});
it("Logging in with Placement Manager Email but Role is Student", (done) =>
{
    chai
        .request(host)
        .get(path)
        .set('content-type', 'application/x-www-form-urlencoded')
        .send({ email: 'placement@daiict.ac.in', password: 'superadmin', role: 'Student'
    })

        .end(function (error, response, body)
        {
            response.should.have.status(500);
            done();
        });
});
});
```

Errored Output:



```
describe('Testing route /login') callback > it('Logging in with Placement Manager Email but Role is Student') callback > end() callback
    .end(function (error, response, body) {
      response.should.have.status(404);
      done();
    });
  });
});
```

17 passing (6s)
1 failing

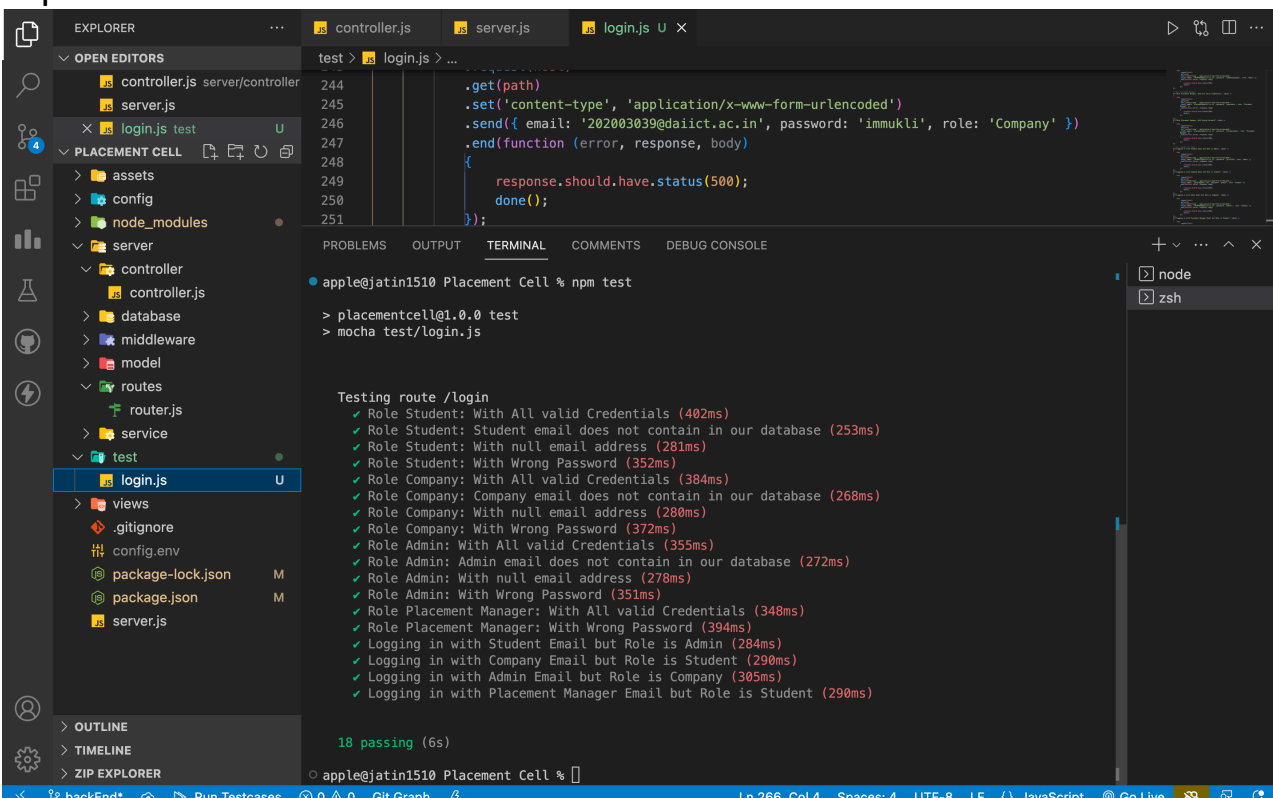
1) Testing route /login
Logging in with Placement Manager Email but Role is Student:

Uncaught AssertionError: expected { Object (_events, _eventsCount, ...) } to have status code 404 but got 500
+ expected - actual
-500
+404

at /Users/apple/Documents/Jatin/DAIIC-Sem6/IT314 - Software Engineering/Project/Placement Cell/I
T314_Project_22/Placement Cell/test/login.js:262:38
at Test.Request.callback (node_modules/superagent/lib/node/index.js:716:12)
at /Users/apple/Documents/Jatin/DAIIC-Sem6/IT314 - Software Engineering/Project/Placement Cell/I
T314_Project_22/Placement Cell/node_modules/superagent/lib/node/index.js:916:18
at IncomingMessage.<anonymous> (node_modules/superagent/lib/node/parsers/json.js:19:7)
at IncomingMessage.emit (node:events:539:35)
at endReadableNT (node:internal/streams/readable:1345:12)
at processTicksAndRejections (node:internal/process/task_queues:83:21)

For the first time, one test case failed. Then we checked our source of failure and noticed that there was a logical mistake for the status code. Then we have corrected it, and we ran the test case again.

Output:



```
test > login.js > ...
    .get(path)
    .set('content-type', 'application/x-www-form-urlencoded')
    .send({ email: '202003039@daiict.ac.in', password: 'immukli', role: 'Company' })
    .end(function (error, response, body) {
      response.should.have.status(500);
      done();
    });
  });
});
```

apple@jatin1510 Placement Cell % npm test

> placementcell@1.0.0 test
> mocha test/login.js

Testing route /login

- ✓ Role Student: With All valid Credentials (402ms)
- ✓ Role Student: Student email does not contain in our database (253ms)
- ✓ Role Student: With null email address (281ms)
- ✓ Role Student: With Wrong Password (352ms)
- ✓ Role Company: With All valid Credentials (384ms)
- ✓ Role Company: Company email does not contain in our database (268ms)
- ✓ Role Company: With null email address (280ms)
- ✓ Role Company: With Wrong Password (372ms)
- ✓ Role Admin: With All valid Credentials (355ms)
- ✓ Role Admin: Admin email does not contain in our database (272ms)
- ✓ Role Admin: With null email address (278ms)
- ✓ Role Admin: With Wrong Password (351ms)
- ✓ Role Placement Manager: With All valid Credentials (348ms)
- ✓ Role Placement Manager: With Wrong Password (394ms)
- ✓ Logging in with Student Email but Role is Admin (284ms)
- ✓ Logging in with Company Email but Role is Student (290ms)
- ✓ Logging in with Admin Email but Role is Company (305ms)
- ✓ Logging in with Placement Manager Email but Role is Student (290ms)

18 passing (6s)