

---

# **Unit Test**

**for**

# **TimeTable Generator**

**CS -08**

**Indian Institute of Information Technology Vadodara**

## **Team Members**

Aman Yadav (201651007)  
DakshKumar Gondaliya (201651014)  
Kirtika Singhal (201651024)  
Mayank Pathela (201651029)  
Nikhil Sachan (201651034)  
Parmeshwar Kumawat (201651035)

## Revision History

Name	Date	Reason For Changes	Version
Parmeshwar Kumawat	Nov 5,2018	Initiate	1
Aman Yadav	Nov 6,2018	Review	1

## Unit Tests

### 1) Testing User Response

```
describe("#GET /users", ()=> {  
  it("Should return an array of users", (done) => {  
    request(app)  
      .get("/users")  
      .expect(404)  
      .expect((res) => {  
        expect(res.body).toEqual(  
          expect.arrayContaining([  
            name : "Aman",  
            age : 20  
          ])  
        )  
      })  
      .end(done)  
  });  
});
```

## 2) Testing Webpage Response

```
describe("# GET /", ()=> {
  it("Should return a response of the webpage" ,(done) =>{
    request(app)
      .get('/')
      .expect(200)
      .expect((res) => {
        expect(res.body).toEqual(
          expect.objectContaining({
            head  : "Test Webpage",
            text  : "This is a webpage to test
express based webapps through supertest"
          })
        )
      })
    .end(done);
  });
});
```

### 3) Testing Authentication

```
describe("GET /users/me", () => {
  it("should return user if authenticated", (done) => {
    var id = users[0]._id;
    request(app)
      .get('/users/me')
      .set('x-auth',
users[0].tokens[0].token, undefined, 2)
      .expect(200)
      .expect((res) => {

expect(res.body._id).toBe(users[0]._id.toHexString());
        expect(res.body.email).toBe(users[0].email)
      })
      .end(done);
  });

  it("should return 401 if not authenticated", (done) => {
    request(app)
      .get("/users/me")
      .expect(401)
      .expect((res) => {
        expect((res.body)).toEqual({});
      })
      .end(done)
  })
});
```

#### 4) Creating New User

```
describe("POST /users", () => {
  it("should create a new user", (done) => {
    var email = "heetHacker@gmail.com";
    var password = "dilonKaCracker";

    request(app)
      .post("/users")
      .send({email,password})
      .expect(200)
      .expect((res) => {

expect(res.headers["x-auth"]).toBeDefined();
        expect(res.body._id).toBeDefined();
        expect(res.body.email).toBe(email)
      })
      .end((err) => {
        if(err){
          return done(err)
        }
        user.findOne({email}).then((user) => {
          expect(user).toBeDefined();

expect(user.password).not.toBe(password);
          done();
        });
      });
    });
  });
});
```