Software Engineering

Session 3

Schedule

Lab

- Node.js and Express.js introduction

Seminar

- Sprint 1 tasks

Recap

- You build a 'CRUD' web application, completely from scratch as a group project
- In the lab sessions, you are taught skills in:
  - **HTML**
  - **Front end Javascript**
  - **Back end Javascript node.js / express**
  - PUG templating engine
  - Connecting your code to a relational database
  - **GIT version control**
  - **Docker environment building/management/deployment**
  - Specification (UML)/ user experience /**project management tools**

You will use these practical skills to build your project.

database

node.js

webserver

MySQL

SQLite

express

Server

response

requests

Browser

HTML
CSS
JS

client

Docker container(s)

Visual studio code (IDE)

git

GitHub

URLS, requests and routes

URLS

'Uniform Resource locator' Where things are on the internet. A unique address consisting of domain name + file path

Request

The data passed in to the webserver to retrieve the information May be GET or POST, uncommonly PUT, DELETE

Route
The mechanism which a webserver uses to understand the request and pass the data to the right code to respond to it

Response
The data passed back to the requesting client.

Routing + requests

This is a get request!!

This is a get request to '/hey'!!

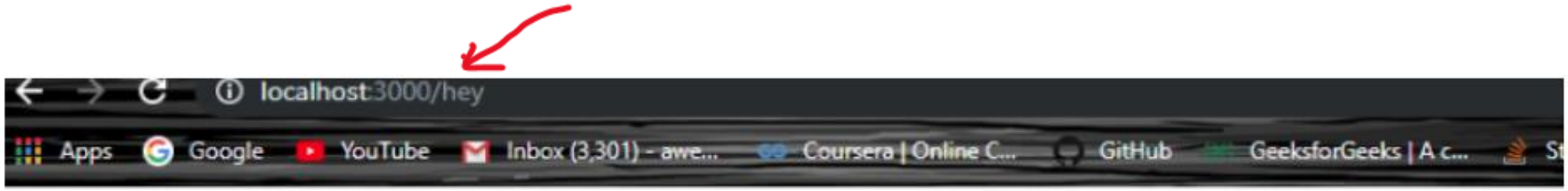
Each **route** responds to a client request to a particular route/endpoint and an HTTP request method (GET, POST, PUT, DELETE, UPDATE and so on).

Each route will refer to different URLs in the website. So when a URL matches a route, the function associated with that specific route is executed

Node.js supplies a simple built in webserver that can receive requests.

Routing

This is a get request!!



This is a get request to '/hey'!!

Key tools and concepts

- Javascript
- Node.js
- npm
- Web server
- URLS
- Express.js
- Routes
- Dynamic routes

Express.js is a module of Node.js

- Provides a more fully functional routing engine

```
app.get('/', (req, res) => {
  res.send('root')
})
```

This route path will match requests to `/about`.

```
app.get('/about', (req, res) => {
  res.send('about')
})
```

This route path will match requests to `/random.text`.

```
app.get('/random.text', (req, res) => {
  res.send('random.text')
})
```

Here are some examples of route paths based on string patterns.

This route path will match `acd` and `abcd`.

```
app.get('/ab?cd', (req, res) => {
  res.send('ab?cd')
})
```

This route path will match `abcd`, `abbcd`, `abbbcd`, and so on.

Express.js dynamic routes

Route parameters

Route parameters are named URL segments that are used to capture the values specified at their position in the URL. The captured values are populated in the `req.params` object, with the name of the route parameter specified in the path as their respective keys.

```
Route path: /users/:userId/books/:bookId
Request URL: http://localhost:3000/users/34/books/8989
req.params: { "userId": "34", "bookId": "8989" }
```

To define routes with route parameters, simply specify the route parameters in the path of the route as shown below.

```
app.get('/users/:userId/books/:bookId', (req, res) => {
  res.send(req.params)
})
```

- Route parameters allow you to build fully dynamic web applications

Lab tasks

- Build your development environment using the scaffolding files
- Create some static and dynamic routes using express.js

BREAK

Sprint 1 prep

Checklist Submission 1
The following must be in place:

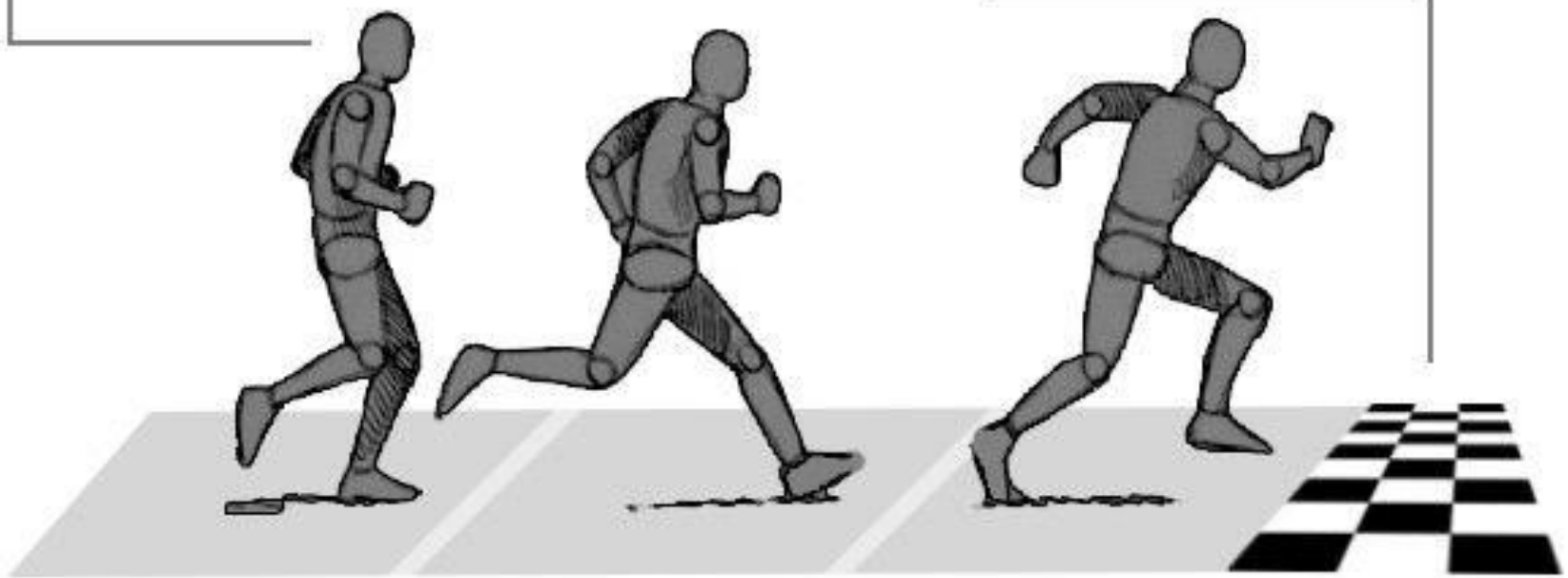
- GitHub repository for coursework set-up and at least one commit from every group member
- Product Backlog created and kanban board in Github project created
- Scaffolding files added to github and README file customised for your project
- Correct branches for GitFlow workflow created - includes `main`, `develop`, and `release` branches
- Each member of the team able to run the development environment using Docker
- A single PDF document with:
    - Your group members and group name
    - Project description refined
    - Code of Conduct
    - At least two ‘Personas’
    - Ethical issues identified
    - Meeting records completed

1. Persona

Defines who the story is about. This main character has attitudes, motivations, goals, and pain points, etc.

3. Goal

Defines what the persona wants or needs to fulfill. The goal is the motivation of why the persona is taking action. When that goal is reached, the scenario ends.



2. Scenario

Defines when, where, and how the story of the persona takes place. The scenario is the narrative that describes how the persona behaves as a sequence of events.

Shraddha Goel

"I want to purchase a comfortable mattress for my baby girl's comfort."

Age: 29 yrs

Occupation: Software Engineer

Status: Married

Location: Delhi

Bio: Shraddha is a Software engineer, mother of 2 yrs. old baby girl. She is responsible to shop for home & she like to try new ways of shopping.

User Story: Shraddha as a mother, wants to find a comfortable mattress for her daughter according to the size of the customized size bed of her daughter.

Goal:

- She wants to give her child the best good night sleep for a healthy and energetic morning.
- She wants to be able to test the mattress for her kid to know the comfort.
- She is Lazy while searching a new place.

Frustration:

- Unable to decide the best material of mattress for her daughter
- Tough to find a place to buy a custom size of mattress to fit her daughter's bed.

User Scenario:

Shraddha is a responsible mother and always look for the comfort of her 2-year-old daughter. From few days she started feeling that her daughter is not getting proper sleep & she believes that mattress might be not that comfortable. She starts searching for a good mattress on her laptop & finds a link to the website that can help in getting a custom size mattress easily and she can also try mattress for few days and if she doesn't find it suitable, she can get it exchanged. She visits the website, searches the mattress & books the mattress to test. She is expecting the mattress in next 24 hour.