Titles

1. Intro to nodejs and backend development

- Syllabus discussion
- Installation : Nodejs, postman , mongodb compass
- what is nodejs -> (nodejs and JS significance)
    - why use nodejs
    - Relation b/w webAPI and nodeAPI -> how these features are accessed
- High level of fs,os,path,http
    - What are the major fetaures that a server requires
        - file system
        - os
        - http
        - path
- Code example different node inbuilt modules (Fs , path , OS)
- Understanding server side development with node (http)-> basic example
- usecase of package.json
- nodemon
- Understanding http module with req and res
- header and body,routes
- Request -> methods ,url & Response -> code , content-type
- Serving static html string, file & json response via the server

2. Intro to API, express and CRUD

- Installation : express ,
- High level of server and API
- High level overview of the lecture
- open api specifications[Postread]
- Intro to express and API development
    - Installation
    - Port and listening
- Res.json
- Request methods and response codes [Pre-read]
- Get request Express
- testing with postman
- Basic Error handling using try and catch
- Middleware by features:
    - Application-level middleware
    - Router-level middleware
    - Error-handling middleware[Post-read]
    - Built-in middleware
    - Third-party middleware
- Middlewares by usecase
    - logging middleware we will be using morgan
    - User defined -> check inputs
    - app.use and 404 page not found
    - POST in express with express.json
    - express.urlencoded([options]):[Post-read]
- Template routes [post-read]
    - getByid [get]
    - UpdateByid [patch]
- DeleteByid [delete]
- PUT, HEADER, OPTION [postread]
- All crud example with the help of fs module[postread]

3. Intro to Mongodb and CRUD operation

- SQL vs NO-SQL[pre_read]
- Intro to DB and it’s usecase
- Overview of the lecture
- How data is stored in Mongodb
  - clusters
    - collection
    - document
    - BSON
    - realworld example
- Setup of Mongodb
  - login
  - creating org
  - creating cluster
  - admin access and
  - network access
- Explaining usecase of env file
  - adding db password and user-name
- Explaining mongodb and mongoose
- Intro to Mongoose -> ODM -> it used to provide schema and model
  - what is schema , model (similar to document)
  - connecting
  - creating model and schema
- CRUD in DB
  - all the queries
  - getProduct, getAllProduct
- All crud example with the help of mongoose module[postread]
- factory Design pattern[postread]

4. Rest API ,Imp API features and integration to frontend

- BMS demo
- showing all the pages
- creating schema for movies
- Restful Design
  - Prinicíples
  - Example
- Creating crud for movies
- Real world use case for pagination, searching, sorting, filtering
- Query params in Express
- Filtering ,searching ,sorting,pagination of movies
  - How mongoose helps in doing these tasks
  - Implementation
- Integrating all these features to frontend
  - making request from react app to express app **Note** : We will need to implement pagination, searching, sorting, filtering on frontend

5. Data validation, aliasing, pre-post hook [postread/recording]

- Data Validation
- pre and post hooks
  - for having custom generes
- Monogdb operator[pre-reads]
- Multiple routes like top 5 products
  - realworld usecase
  - how it is implemented in backend
- Integrating the above features in frontend
- error handler wrapper [postread]

6. Login Signup,Hashing Password

- creating userModel
- storing password
  - caveats[postread]
  - different hashing algorithms [postread]
  - why bcrypt is better-> explination of bcrypt algo [postread]
  - using pre-hook to remove confirmPassword and hashing of the password
- UI for login and signup, Profile Page
- Redux flow -> auth

7. Authentication,JWT and cookies

- cookies as transport mechanism
  - cookies history , intution
  - usecase of cookies
  - impact of cookies in day to day life
  - working with cookies in express
- Authentication ,authorization and identification
- Usecase of authentication
- JWT algorithm
  - structure
  - how creation happens
  - how verification happens
- using cookies to send JWT token
- Validations [Postread]
- Protect route middleware
- Login with token + localstorage[postread]
- Profile Page
- logout [Postread] ## 8. Update,reset,forgot Password & sending email
- SMTP server and Emails [postread]
- Sending Email using sendgrid and Nodemailer
  - differente components required to send and email programatically
  - Limitation of styling in css with email
  - signup to sendgrid
  - creating API key
  - sending email with a template
- Implementing routes in backend
  - Update password
  - Forget password
  - Reset password
- Redirects in express
- Integrating email sent on forget password
- creating UI for Forget and Reset [Preread].

9. Authorization , conditional flow of data , Admin dashboard

- What is Authorization and it’s usecase
- Ways to implement Authorization
  - static version
  - dynamic version [postread]
- Integrate that authorization to our backend
- Implementing dynamic roles using pre hook [postread]
- Integrating authorization to frontend

10. Data Modelling and Theatre API and MVC

- converting the code into MVC architecture
  - MVC architecture explanation

    - Converting to MVC
    - Mounting
    - Creating Routes
    - Createing controllers
    - Chaining handlers
- Build the Theater API
- User can apply for Theater and Admin has the right to approve it , then theatre will be added

11. Payment Gateway Integration and shows API

- Integrate Stripe to Make payment
    - ways to do payment
    - usecase of payment gateway
    - flow of payment
    - stripe setup
    - payment poc
    - Features of payment gateway[postread]
- Build the seat booking API and and
- Build the Shows API that will run in different theatres
- Build the shows components which will show the Show Details with name of theater and movies
- Make Seating Arrangement Component
- ngrok and tunneling and webhooks
- Generate Ticket
- Doing payment for the movie

12: security.

- Security
    - Guidelines for security policy of your backend
- Ways to secure
    - Password hashing
    - Adding rate limiter
    - Adding required headers ->
    - on your own
    - using helmet
- Preventing XSS
- NoSQL injection

13 : Testing, CORS,CSRF and Deployment []

- What is cors
- How to solve cors
- session managemnt , CSRF and it’s mitigation
- Deploy MERN stack app to render
- Testing in backend[postread]

14 : Websockets

- usecase of realtime applications
- showing the comm in dev tools
- HOw HTTP works
- How websocket are a better at real time communication
- Process of upgradation from HTTP -> websockets
- Websockets using node js
    - creating express server
    - sending event
    - recieveing
    - sening to one guy

    - sending to multiple peole
- Web sockets
    - features
    - pros
    - cons
- Project : Chat application ## 15 : Backend Interview Prep-1
- Lec-1 [preread]
- Start with discussion of Agenda
- [Wow factor] : Show two codes files
    - problem related to large file sizes
    - Problem related to cpu heavy processes.
- Types of modules in nodejs
    - inbuilt modules
    - local modules
    - third party module
- Nodejs modules
    - fs examples
    - os examples
    - path examples
- Nodejs streams
    - Showing the problem again
    - Introducing streams
        - types of streams
            - there examples
    - solving the problem
- Nodejs child process
    - Showing the problem code with fib
    - Explaining all the methods of child process
    - spawn and opening chrome
    - fork and solving our problem ## 16: Backend Interview Prep-2
- review of last lecture [preread]
- first principles of Nodejs
    - event driven
    - single threaded
    - asynchronous
- Event Emitter and custom Event Listener
- custom event listenere(observer design pattern)[postread]
- How require works in node js [postread]
- Http server and testing single threaded nature of server
- Understanding architecture of nodejs ->
    - high level
    - low level Identifying things that are async and those are sync
- threads , processes and nodejs [postread]

Extra topics

- session management
- CSRF token[postread]
- Migration script
- central error handler
- Builder design pattern [postread]

Take home Projects

- Streaming server