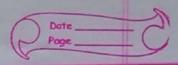
## THEORY ASSIGNMENT - 1

NAME - KRISHI TAILOR

ROU NO - 75

SEMESTER - 7th

SUBJECT - APPLICATION DEVELOPMENT USING FULL STACK

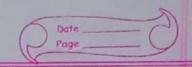


Node je - Intuoduction, features, execution Q.1 auchitective. Node je is a cuoss platform suntime emisionment Any and library for running javasuriet applications outside the business. It is used for meating Suwur- orde and networking web application. It is open-source and free to use- many of the basic modules of moders are written in javascript. Noch je is mostly used to sum real time survey applications. Node js also provides a sich liberary of various javaswipt module to simplify the dwillogment of web applications. Node-js = Runtine Environment + Javasuript library Features of Node js: Extremely fast: Noch js is built on google duronie's 18 janascript engine, so its library is very fast in code execution. In Single threaded: Node js follows a single threaded model with went looping.

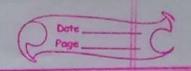
Highly Scalable: Nocle js is lughly scalable because mechanism hups the survey to suspond in non-stocking way.

\* Lieuse:- Node js is released under the

\* Open Source: Noch js has an open source community which has produced many exculent modules to add additional capabilities to node js application.

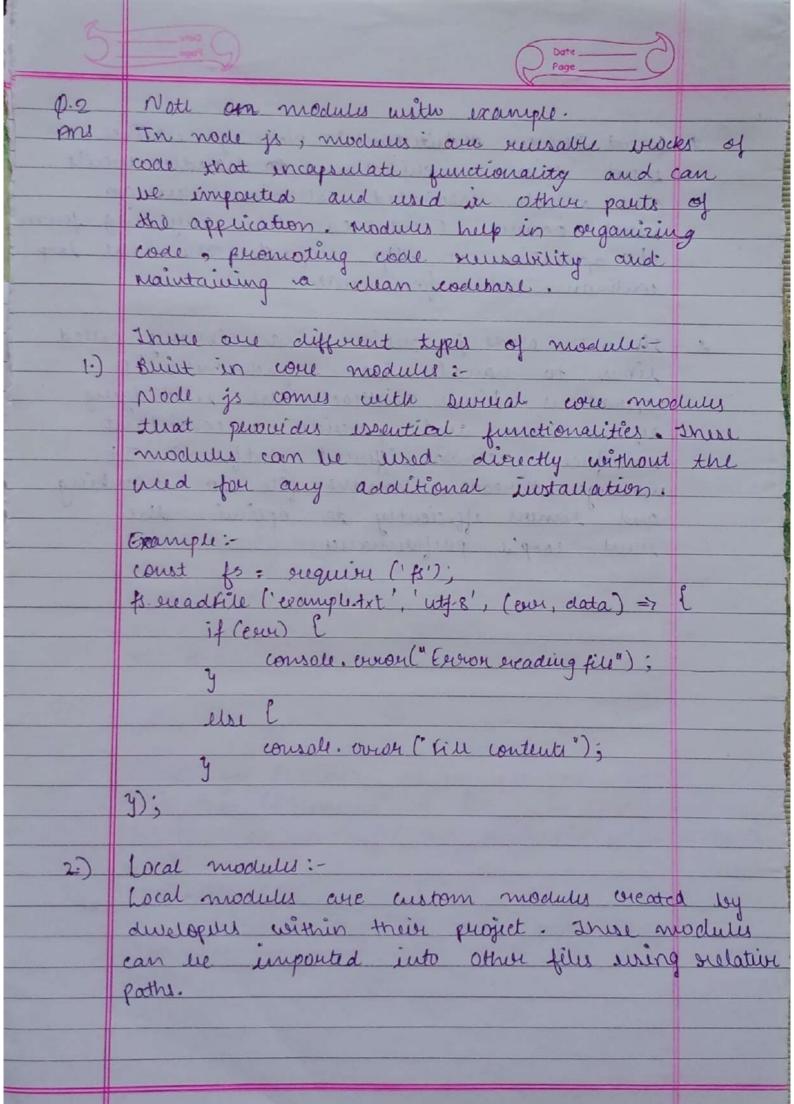


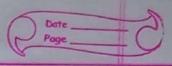
- \* No enforing: Node js cuts down the overall processing time while uploading audio and video files. Node js doesn't buffer any data
- 1/0 is Agnamonous: All ATI of mode is library are asynchronous i.e., non-blocking. So, node is brasid source never waits for an ATI to return data.
- Event loop: The west loop is susponsible for draudling asynchronous operation and 110 task efficiently. In keeps an eye on the message queue and executes the can back functions associated with completed 110 operations when the can stack is empty.
- I Single threaded, Non Blocking: Node is operated on a single threaded event loop. This might seem countrientuitive for handling mutiple concurrent eliquests; but mode is activer this through non blocking 110 operations. When a sequest everywhere are 110 operation, node is deligated the task to the septem.
- \* Compacks and Psynchronous Programming:Callbacks play a significant scole in givele js
  instead of waiting for a function to
  compute before moving to the net one,
  nocle js uses caubacks to notify the
  completion of an asynchronous task.



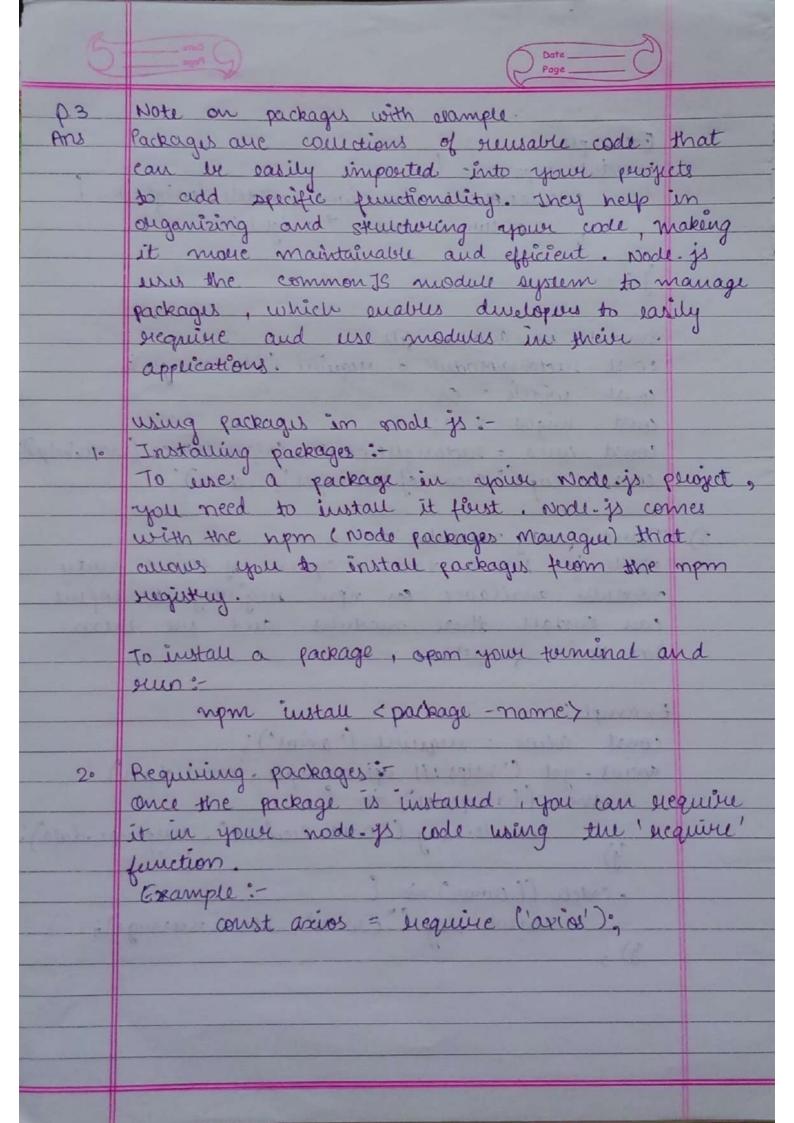
\* Event Denvers Auditecture: Node js utilizes an went drivers curchitecture to handle wents and execute associated callbacks when an went occurs. Event can be anything from 110 operation to Http suguests. The went loop continous listens for wents.

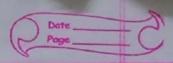
Libur: - Node js seelies on a library called libur to handles its asynchronous 110 operations. Librar abstracts the undulying septem carry and provides a consistent interferce accross different platforms. It manages threads, handles file 110 networking and times efficiently to optimize the west roop's performance.





Example:module exports = 6 calculations (width, neight) outurn width \* neight; module in another fileconst endangerodule : réquire l'Indangers const width = 5; const unea = suctangle Module. Calculationerea (wiath, height); consoli. 109 (" Auea of endanger", area); 3) Third party modules: Node je has a vast ecosystem of third party anduly available on rem registry. Developers can enstall these modules and use them In their projects. Escampli: const actios = nequire ('acrios'); accios. get ('https:// api. example.com/ data) -then ((suspense) => ( console. log ("Data Received", susponse.data); · catch ((cown) => ( console. log ("Error", eteror. nussage);





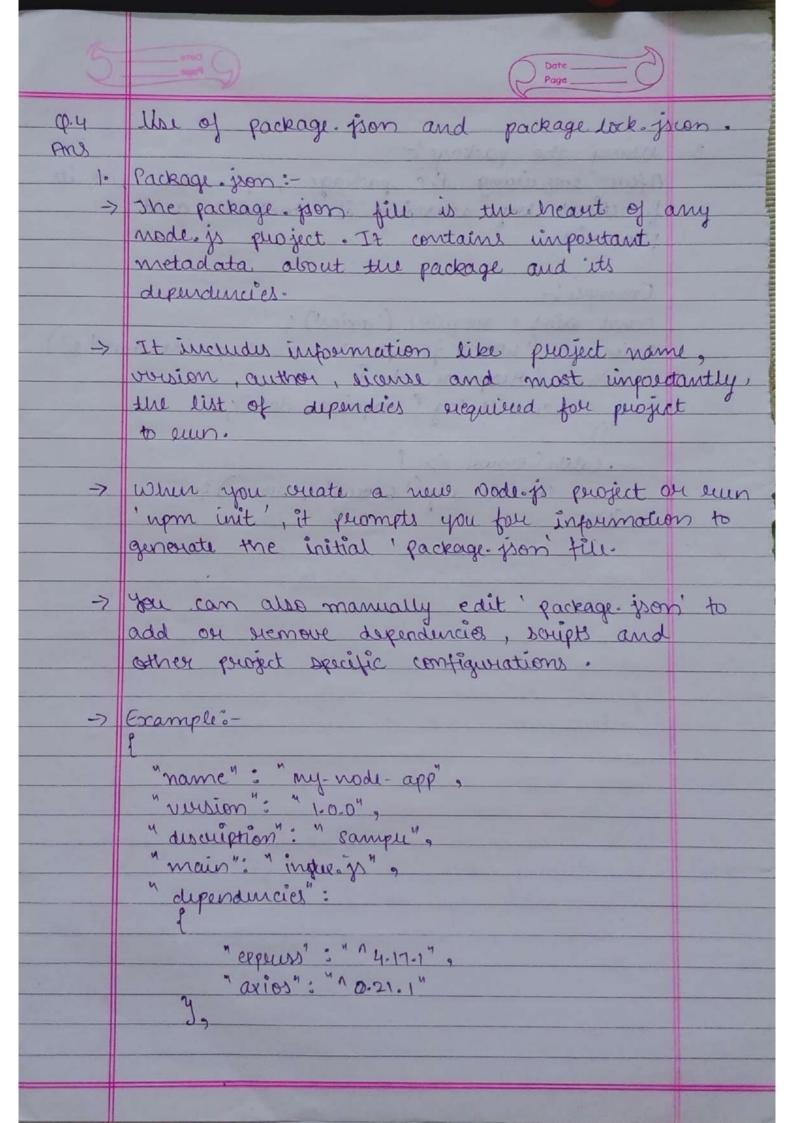
3. Using the package:

After sequiring the package, you can use its

functions by accessing its methods and proporties. Example:const aros = require ('axios'); aries. get ("http:// jsomplaceholder. typecode.com/1")

Then (rusponse => {

console.log (rusponse.data); · catch (course >) { 3); console engl'Esonor!, esonor);

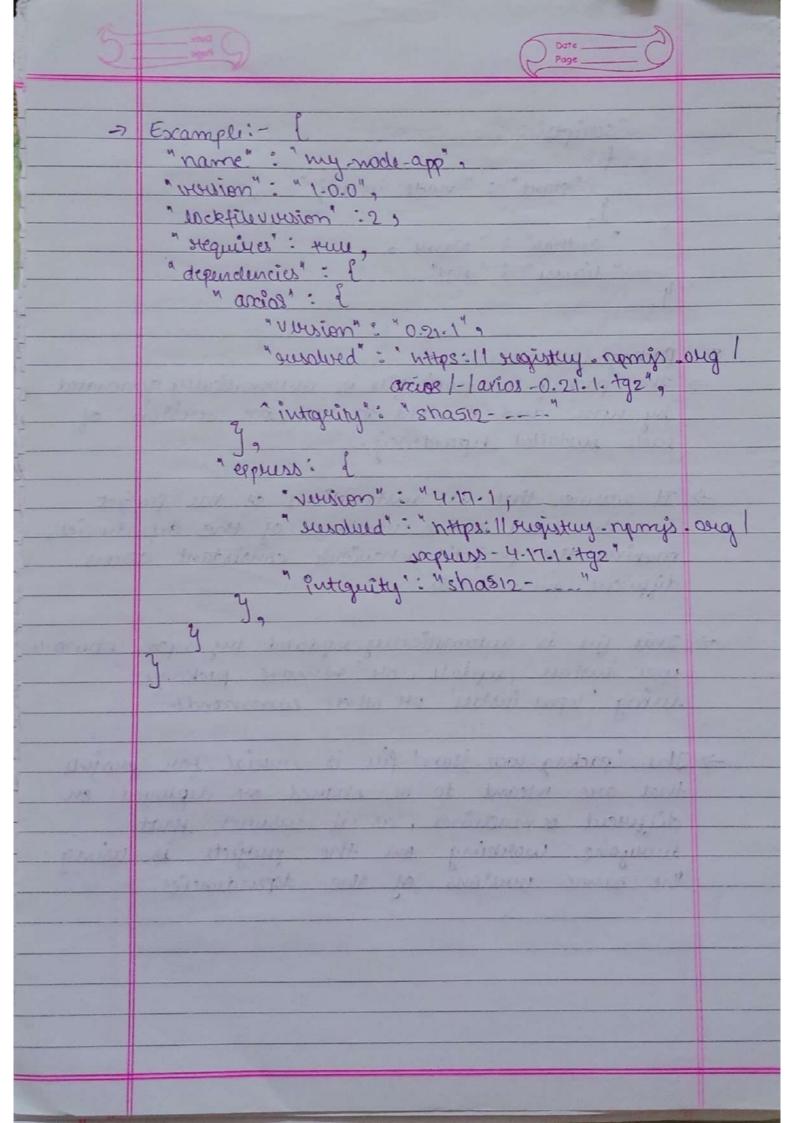


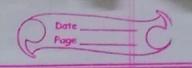


"start": "node inder-js"

"authori": "Name",
"liceuse": "MIT"

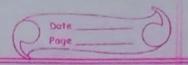
- 2. Package-lock-json:
  - > The package-lock from the is automatically generated by upon and is used to lock the version of each installed dependency.
  - -> It ensures that all instantions of the project use the exact same vorsions of the dependencies, making the project's behaviour consistant across different eminenments.
- -> This file is cutematically updated by non whenever you install, update, or survive packages Lising 'nom install' or other commands.
- -> The 'packag-wock. json' five is crucial for projects
  that are meant to be shared or deployed on
  different or machines, as it ensures that
  ensurement working on the projects is using
  the same versions of the dependencies.



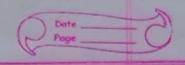


PMS In the Node-15 ecosyptem, packages are collection of modules, libraries and ecosyrus that dwelopies can use to enhance their applications.

- Popular web frameworks that simplify the process of brilding web applications and APIS by providing storting, middle coare supposit and other features.
- 2. utility libraries: Packages like lodash, Romda and surdusure is privide estility functions that assist with tests like data manipulation, validation and functional programming.
- 3 Potabasi libraries: Packages like Mongoose and sequelize provide easy-to-use abstractions for working with dotabases and orm (object selation marking) capabilities.
- 4. Authentication and Security: Packages sike possepost-js often solutions for authentication and password nashing to enhance application security.
- 5. Template origines: Packages like EJS, Pug and brandlebours smable durlopers to generate dynamic HTML content easily.



- 6- HTTP clients: Packages like Axios and sequest provide took for making HTTP sequest, allowing vodijs applications to sutreact with APIs and web ruwices.
- T. CLI (command line) tools: Parkages like commander and years emable duelopers to will into active and user-friendly command line tools.
- 8. Real-time sommunication: Packages like socket, ID facilitate great-time communication between clients and sower using web sockets.
- 9. File system stilities: Packages like fs: evera and glob provide additional functionalities and east of use for working with file septem.
- 10. Date 1 Time manipulation: Packages like moment-is and see-is offer tooks for parsing, formatting and manipulating dates and times.
- 11. lagging: Packages like winston and Bungan puovide flexible lagging solutions for node js applications.



P.6 npm introduction and commands with its

ANS.

is a command-line tool and the default package manager for node of It simplifies the praces of installing, updating, and managing node is packages and their dependencies, now also provide a central reportation where duelopers can publish and shall wheir node is packages, making it shall be their node is packages, making it shall be their node is packages, making it say for others to use and contribute to their code.

remmands with its usi
I Tritializing a new node is project.

To create a new node is project, you can

use the 'upm init' command. It will quide

you through a socies of prompts to create

a 'package ison' file, which contains information

about the project.

~>> npm init

2. Installing packages:

To install packages for your made is project,

Lust the 'npm install command tollowed by

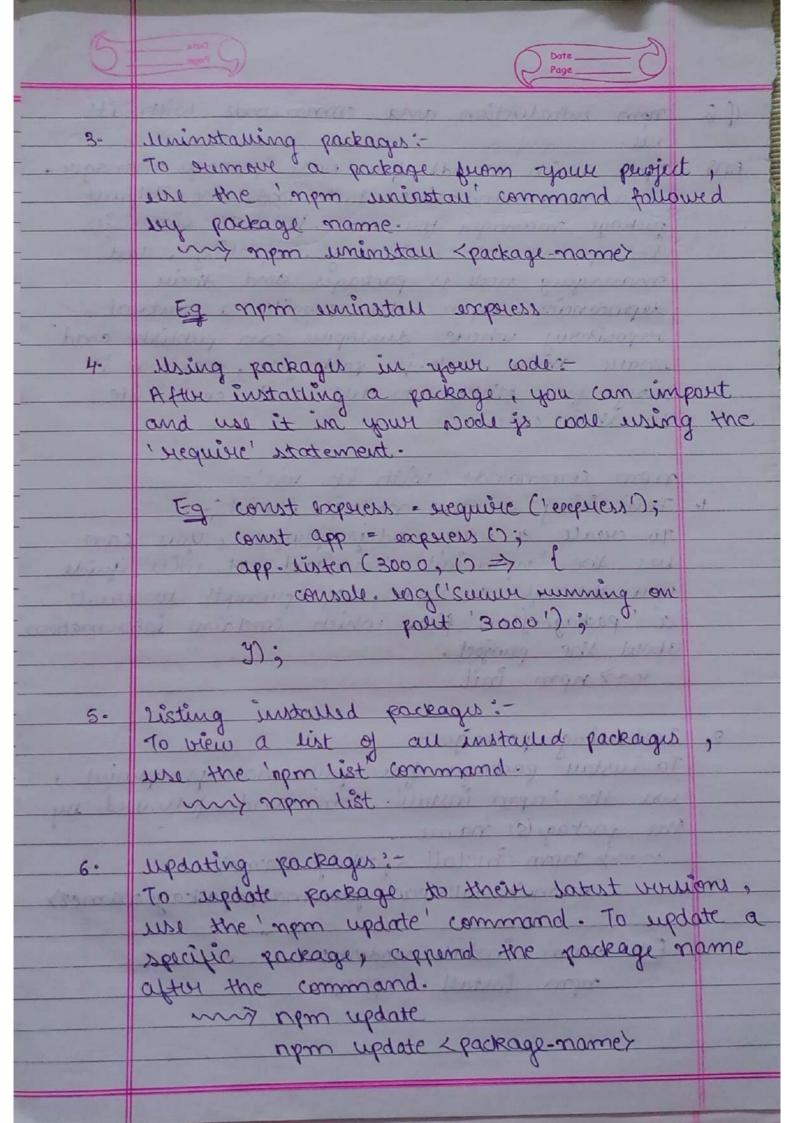
the yockage(s) name.

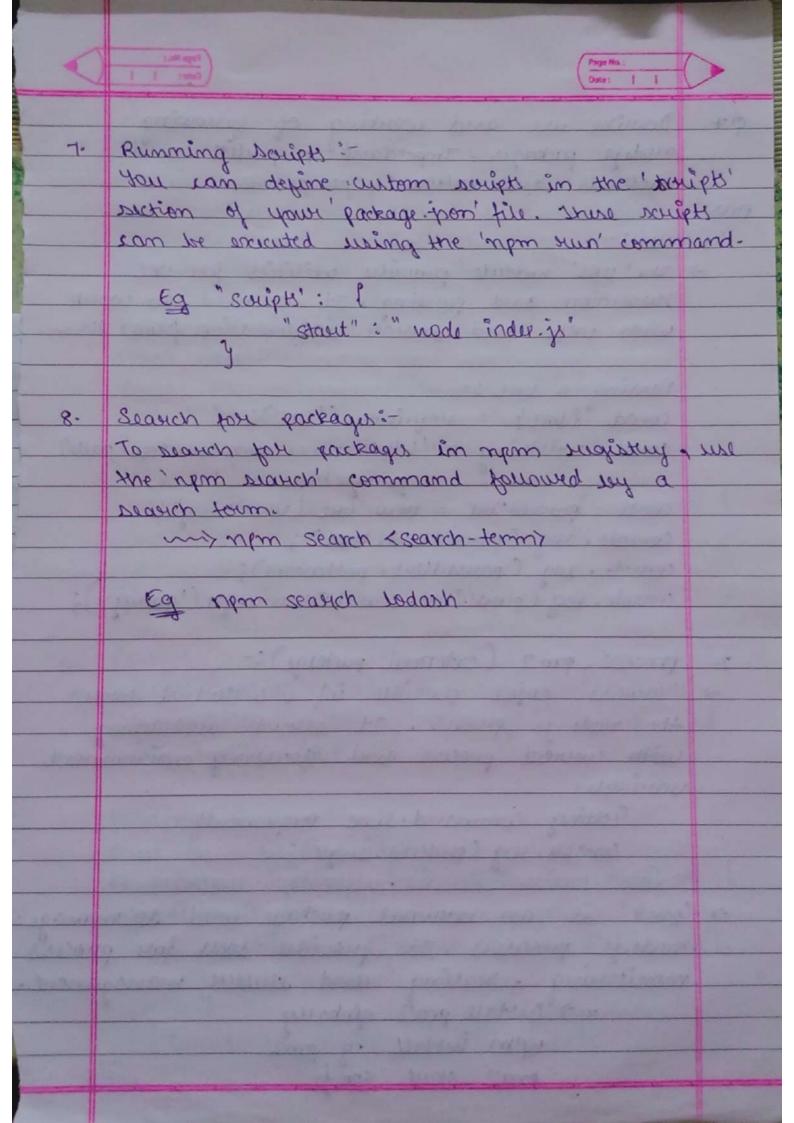
My npm install <package-name>

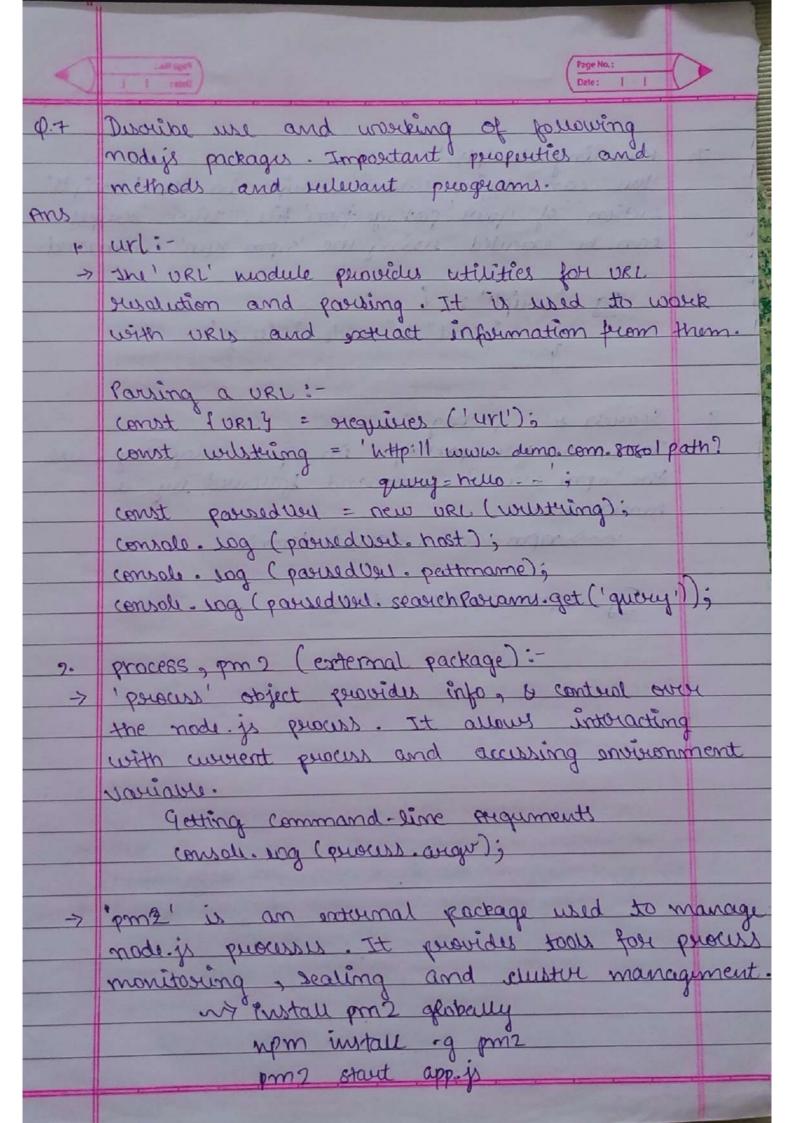
npm install <package-name>

npm install <package-name>

Eg nom install nodemen







3. readline :--> The 'Headline' module provides an interface for reading input streams line by line. It is ni visus ative to interact with issens in the command line anvivenment. Reading issu input: court readline = require ('readline); court sil = readline : create Interface !! input: process. stding; output: persons. stdout 3 el question (' what's your name? ', ( name) => courale. log ('Hello, \$ {name y!'); su close (); 3); 40 fs :--> The 'fs' module provides file system - evelated functionality, allowing reading, writing and manipulating files Reading a file :court is = evequine ('xs'); fs. sreadfile ('example.txt'3, 'utf-8', (cros, data) => if (croi) of course every (" Everer exading file", ever);

console. log (" File content: ', data);

5. events:--> The 'events' modules provides on event-driven auchitecture for building applications that can emit and listen to events. const EventEncitor = require ('events'); class ryEncitor extends & organia (Constable) ? y const myEmitter = new my Emitter (); my Emitter on ('quiet', (name) => [ consoliting ('Hello, framey'); my Emitter. emit ('greet', 'Domme'); 6. Console: -> The 'compole' module pravides a simple debugging console that can be used to log messages during development. consolillog ('This is a log message'); console error ('This is an owner message'); coursel. tenera ('This is waving merrage'); 7. buffer: -> The buffer module priorides a way to handle timouy data. It is used to work with seaw simony data in node is applications. var but = buffer. ferom ('abc'); (ansole. log (but);

8- querystring:--> The 'quingstring' module provides utilities for working with query string in URLS const querystring = require ('querystring'); const params = querystring. pares ('name = asi & cege=10 consoli. Log (params); 9. http:-> The 'http' module provides a set of functions and classes to weate HTTP survers and make HTTP sequests. const http = stequire ('http'); const runer = http. vieatiserver ( sieg, vier) => ( Her writefread (200), ( Content -Type': 'text' 3) sus. end ('Hello', 'would'); Seewed - listen (3000, 1) > { console eg ('Seewer is running on port 3000' -> The 'V8' module exposes APIs helated to the vo javascript engine, providing access to performance and memory related data. const v8 = siequire ('v8'); console. sog (ve-getteapStatisticsU);

> The 'os' module perouides operating system related function acity such as information about the nost operating system. const os = require ('os'); console. eg('os platform', os. platform (); conside. eng ('CPU recluitecture", os- auch (1); 12. 2lib: -> The 'slib' module provided compression and decompussion functionalities using grip and deplate. > compensing and decompulsing const slib = stequise ('slib'); const data = "This is some data to compless"; zlib. grip (data, vur, comprussed data) > couple surge ("Compussion eval, ever); sieturin; console sog ('comprused data', comprused Pata); zlib unzip (comprused pata, Cerus, duemprussed Data) => consoli. every (" Decomposaned every ¿ mentere

