**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **25 MAY 2020** | **Name:** | **MANAVI** |
| **Course:** | **Signals and system** | **USN:** | **4AL18EC031** |
| **Topic:** | **Day 1: Introduction to Fourier Series & Fourier transform**  **Fourier Series – Part 1**  **Fourier Series – Part 2**  **Inner Product in Hilbert Transform**  **Complex Fourier Series**  **Fourier Series using Matlab**  **(Use Octave to execute the code)**  **Fourier Series using Python**  **(Experience implementation using Python**  **Fourier Series and Gibbs Phenomena Using python** | **Semester & Section:** | **4TH SEM**  **& A SEC** |
| **Github Repository:** | **Manavi-test** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session** |
| **Report – Report can be typed or hand written for up to two pages.**  **FOURIER SERIES AND FOURIER TRANSFORM:**-   * early in the 19th century fourier studied sound and oscillatory motions and conceived of the idea of representing periodic functions by their coefficient in an expression as a sum of sines and cosines rather than their values. * he noticed, for example, that you can represent the shape of a vibrating string of length L, fixed at ends, as * (observe that indeed the coefficients contain important and useful information about the quality of the sound that the string produces that is not easily accessible * from the ordinary description of the shape of the string. * this kind of representation is called a fourier series and there is tremendous amount of mathematical lore about properties * of such series and for what classes of functions they can be shown to exist * one particularly useful fact about them is the orthogonality property of sines * for nonnegative integers j and k here is the kronecker delta function which is 0 if j k and 1 if * the integers j and k here is the kronecker delta function which is 0 if j k and 1 if j= k * the integral above then is 0 unless j k each of these sines as half of these cosines integrates to 0 over this range * by multiplying the expression for y (x) above by sin and integrating the result from 0 to L by the orthogonality property everything cancels except the sin term and we get the expression      |  |  |  |  | | --- | --- | --- | --- | | **DATE:** | **25 MAY 2020** | **NAME:** | **MANAVI** | | **COURSE:** | **PYTHON** | **USN:** | **4AL18EC031** | | **TOPIC:** | **application 4:build personal website with python and flask** | **SEMESTER & SECTION:** | **4TH SEM & A SEC** | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | |
|  |  |  |  | |
|  |  |  |  | |
|  |  |  |  | |
| **AFTERNOON SESSION DETAILS** | | | |
| **Image of session** | | | |
| **Report – Report can be typed or hand written for up to two pages**  **BUILD PERSONAL WEBSITE WITH PYTHON AND FLASK:-**   * the general idea is to create a simple myapp.py which should look somethin like this: * you also want to create a very basic home.html template. * you can make it fancier ie pass variables via render\_template and use them in the html file,reference and external css file instant of some controversial in-line css,and so on. * however,if you go a head and run python myapp.py,you should see that the web site is up and running on your local host- if this is . * if you think once again about it,what we want to build is somewhere to display our bio and some links,and really nothing more. * in other words,our end goal is to make a static website,and we can leverage frozen-flask to freeze a flask application into a set of static files. * frozen-flask is only about deployment instant of installing python and flask on your server,you can use frox=zen -,flask to freez your application and only have static html files on your server. * its magic (ok not really magic ,but thats quite impressive,right?) * i found this article quite clera that helpful,but in a nutshell ,after installing the frozen -flask. | | | |