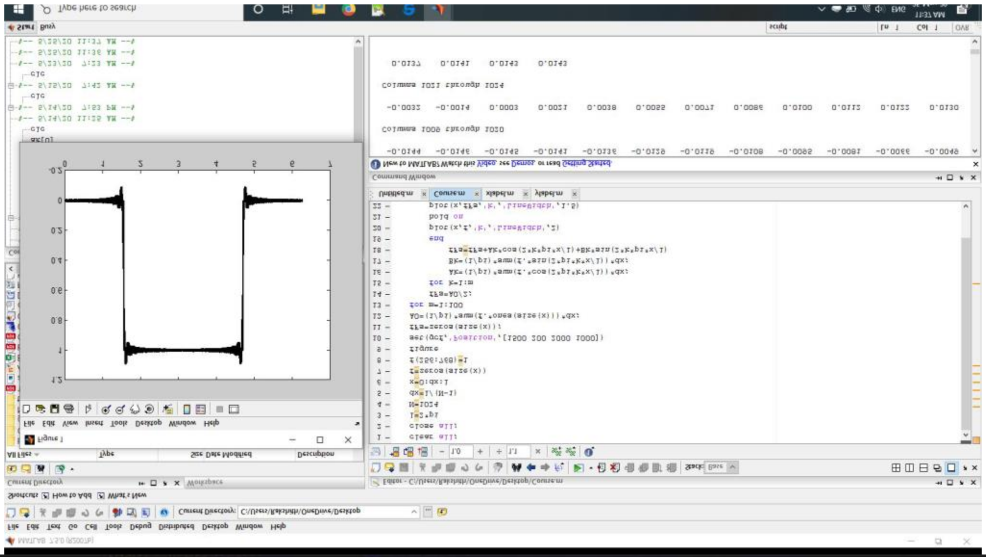
DAILY ASSESSMENT FORMAT

Date:	**25/5/2020**	**Name:**	**Kishan shetty**
Course:	**TCS ION**	**USN:**	**4AL17EC003**
Topic:	**Introduction to Fourier Series, Fourier transform, Hilbert Transform, Fourier Series Using Matlab**	**Semester & Section :**	**6th-‘A’**
GitHub Repository:	**Kishanshetty-041**	**E-mail:**	**Shettykishan983@gmail.com**

FORENOON SESSION DETAILS

Image of session



```
clear all;
close all;
L=2*pi
N=1024
dx=L/(N-1)
x=0:dx:L
f=zeros(size(x))
f(256:768)=1
figure
set(gcf,'Position',[1500 200 2000 1000])
fFS=zeros(size(x));
A0=(1/pi)*sum(f.*ones(size(x)))*dx;
for m=1:100
    fFS=A0/2;
    for k=1:m
        Ak=(1/pi)*sum(f.*cos(2*pi*k*x/L))*dx;
        Bk=(1/pi)*sum(f.*sin(2*pi*k*x/L))*dx;
        fFS=fFS+Ak*cos(2*k*pi*x/L)+Bk*sin(2*k*pi*x/L)
    end
    plot(x,f,'k','LineWidth',2)
    hold on
    plot(x,fFS,'r','LineWidth',1.5)
```

```
  -0.0144  -0.0146  -0.0145  -0.0141  -0.0135  -0.0126  -0.0116  -0.0103  -0.0089  -0.0074  -0.0059  -0.0046

  Columns 1009 through 1020

  -0.0032  -0.0014   0.0003   0.0021   0.0038   0.0055   0.0071   0.0086   0.0100   0.0112   0.0122   0.0130

  Columns 1021 through 1024

   0.0137   0.0141   0.0143   0.0143
```

Figure 1
File Edit View Insert Tools Desktop Window Help
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Untitled.m
Course.m
xlabel.m
ylabel.m
Editor - C:/Users/Rakshith/OneDrive/Desktop/Course.m
Current Directory
Workspace
script
Ln 1
Col 1
OVR
11:37 AM
ENG

databookuw.com
Gibbs
$f(x) = \sum a_k \cos\left(k\frac{2\pi x}{L}\right) + b_k \sin\left(k\frac{2\pi x}{L}\right)$
$a_k = \langle f(x), \cos\left(k\frac{2\pi x}{L}\right)\rangle$
$b_k = \langle f(x), \sin\left(k\frac{2\pi x}{L}\right)\rangle$

Fourrier Series Using Matlab

```
clear all
close all
clc
figure
set(gcf,'Position',[1500 200 2000 1200])

%define domain
L=pi;
N=1024;
dx=2*L/(N-1);
x=L:dx:L;

%Define hat function
f=0*x;
f(N/4:N/2)=4*(1:N/4+1)/N;
f(N/2+1:3*N/4)=1-4*(0:N/4-1)/N;
plot(x,f,'-k','Linewidth',3.5),hold on

%compute fourier series
CC=jet(20)
A0=sum(f.*ones(size(x)))*dx/pi;
fFs=A0/2;
for k=1:20;
A(k)=sum(f.*cos(pi*k*x/L))*dx/pi;
B(k)=sum(f.*sin(pi*k*x/L))*dx/pi;
fFs=fFs+A(k)*cos(k*pi*x/L)+B(k)*sin(k*pi*x/L);
plot(x,fFs,'-','color',CC(k,:),'Linewidth',2)
pause(.1)
end

%% plot amplitudes
figure;
set(gcf,'Position',[1500 200 2000 1200])
clear ERR
clear A
fFs=A0/2;
A(1)=A0/2/pi;
ERR(1)=norm(f-fFs);
kmax=100;
for k=1:kmax
A(k+1)=sum(f.*cos(pi*k*x/L))*dx;
B(k+1)=sum(f.*sin(pi*k*x/L))*dx;
fFs=fFs+A(k+1)*cos(k*pi*x/L)+B(k+1)*sin(k*pi*x/L);
ERR(k+1)=norm(f-fFs)/norm(f);
```

```
end
thresh=median(ERR)*sqrt(kmax)*4/sqrt(3);
r=max(find(ERR>thresh));
r=7;
subplot(2,1,1)
semilogy(0:1:kmax,A,'k','linewidth',1.5)
hold on
semilogy(r,A(r+1),'co','Linewidth',15,'MarkerFaceColor','c')
xlim([0 kmax])
xlim([10^(-7) 1])
ylabel('Mode Amplitude','FonSize',16)
subplot(2,1,2)
semilogy(0:1:kmax,ERR,'k','Linewidth',1.5)
hold on
semilogy(r,ERR(r+1),'co','Linewidth',15,'MarkerFaceColor','c')
xlabel('Mode Number,k','FontSize',16)
ylabel('Reconstruction Error','FontSize',16)
```

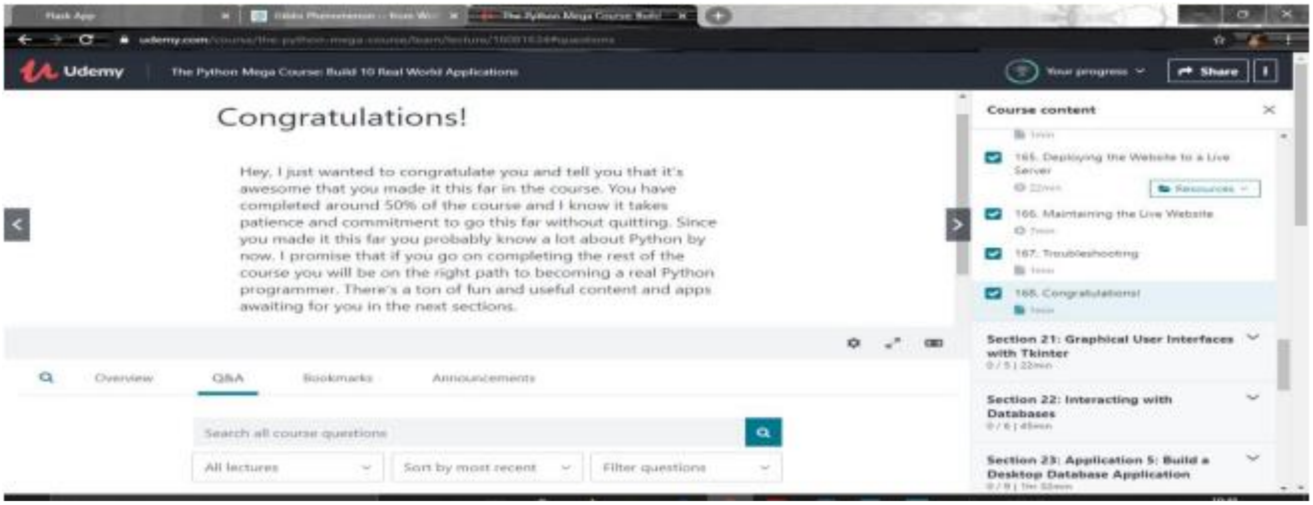
Fourier Series and Gibbs Phenomena [Matlab]

```
clear all;
close all;
l=2*pi
N=1024
dx=l/(N-1)
x=0:dx:l
f=zeros(size(x))
f(256:768)=1
figure
set(gcf,'Position',[1500 200 2000 1000])
fFs=zeros(size(x));
A0=(1/pi)*sum(f.*ones(size(x)))*dx;
for m=1:100
fFs=A0/2;
for k=1:m
Ak=(1/pi)*sum(f.*cos(2*pi*k*x/l))*dx;
Bk=(1/pi)*sum(f.*sin(2*pi*k*x/l))*dx;
fFs=fFs+Ak*cos(2*k*pi*x/l)+Bk*sin(2*k*pi*x/l)
end
plot(x,f,'k','LineWidth',2)
hold on
plot(x,fFs,'k','LineWidth',1.5)
pause(0.1)
end
```

Date:	**25/5/2020**	**Name:**	**Kishan shetty**
Course:	**UDEMY-The Python Mega Course: Build 10 real world applications**	**USN:**	**4AL17EC041**
Topic:	**Application 4: Build a Personal Website with Python and Flask**	**Semester & Section:**	**6th-'A'**

AFTERNOON SESSION DETAILS

Image of session


Flask App
The Python Mega Course: Build
udemy.com/course/the-python-mega-course/learn/lecture/10081524#questions
Udemy
The Python Mega Course: Build 10 Real World Applications
Your progress
Share
Overview
Q&A
Bookmarks
Announcements
Search all course questions
All lectures
Sort by most recent
Filter questions

Congratulations!

Hey, I just wanted to congratulate you and tell you that it's awesome that you made it this far in the course. You have completed around 50% of the course and I know it takes patience and commitment to go this far without quitting. Since you made it this far you probably know a lot about Python by now. I promise that if you go on completing the rest of the course you will be on the right path to becoming a real Python programmer. There's a ton of fun and useful content and apps awaiting for you in the next sections.

Course content

1min

☑ 165. Deploying the Website to a Live Server
22min Resources

☑ 166. Maintaining the Live Website
7min

☑ 167. Troubleshooting
1min

☑ 168. Congratulations!
1min

Section 21: Graphical User Interfaces with Tkinter
0 / 5 | 22min

Section 22: Interacting with Databases
0 / 6 | 45min

Section 23: Application 5: Build a Desktop Database Application
0 / 9 | 1hr 32min

Report – Report can be typed or hand written for up to two pages.

Application 4: Build a Personal Website with Python and Flask

- A local Website can be created using Flask package under flask library in just seven lines of code.
- The html files which are required to design the webpage are saved in the folder named templates.
- The html files home.html and about.html files (child template) are linked to layout.html (parent template) file for navigation menu using extends tag.
- For css styling of the webpage main.css file is created under the folder static\css.
- To deploy the web app into a live server, Git software is used. Git is a version control system allowing you to upload the project files to a server and helps track your changes while maintaining the web app.
- Steps to deploy a static Flask website to Heroku:

1. Create an account on www.heroku.com, if you have one already then login to Heroku.
2. Download and install Heroku Toolbelt from https://devcenter.heroku.com/articles/heroku-cli.
3. Install gunicorn with "pip install gunicorn", gunicorn is a http server which Heroku needs to run web application.
4. Create a virtual environment in python using virtualenv package.
5. Create a requirement.txt file in the main app directory where the main Python app file is located. You can create that file by running "pip freeze > requirements.txt" in the command line. Make sure you're using pip from your virtual environment if you have one. The requirement.txt file should now contain a list of Python packages.
6. Create a file named "Procfile" in the main app directory. The file should not contain any extension. Then type in this line inside: "web: gunicorn app4:app" where "app4" should be replaced with the name of your Python script and "app" with the name of the variable holding your Flask app.
7. Create a runtime.txt file in the main app directory and type "python-3.7.7" inside. By default, Heroku takes python-3.6.10 into consideration.
8. Open your computer terminal/command line to point to the directory where the Python file containing your app code is located.
9. Using the terminal, log in to Heroku with command "heroku login"
10. Enter your Heroku email address and password. 11. Create a new Heroku app with "heroku create my_app_name"

12. Initialize a local git repository with "git init"

13. Add your local application files to git with "git add ."

14. Tell git your email address with "git config --global user.email "myemail@gmail.com"". Make sure the email address is inside quotes here.

15. Tell git your username (just pick whatever username) with "git config --global user.name "what_ever_username"". The username should be in quotes.

16. Commit the changes with "git commit -m "first commit"". Make sure "first commit" is inside quotes.

17. Before pushing the changes to Heroku, tell Heroku the name of the app you want to use with "heroku git:remote --app my_app_name" 18. Push the changes to Heroku with "git push heroku master"

19. Open your app with "heroku open" command.

- After deploying the web app to Heroku, when you visit the website on the browser you see an error, probably something went wrong during the deployment.
- You can see what you what went wrong during deployment by looking at the server logs. You can access the server logs by running "heroku logs" command in the terminal.
- This command will show a series of messages. Carefully read the logs to understand what went wrong.
- If there are any future changes to made (maintenance) can be completed with the help of git software.