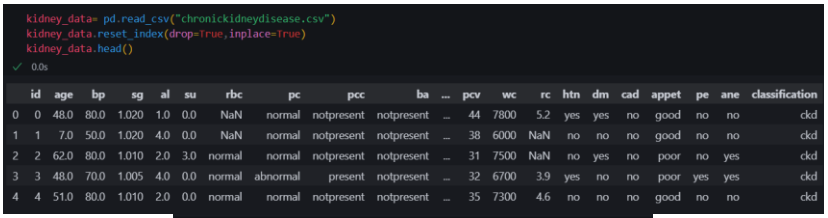
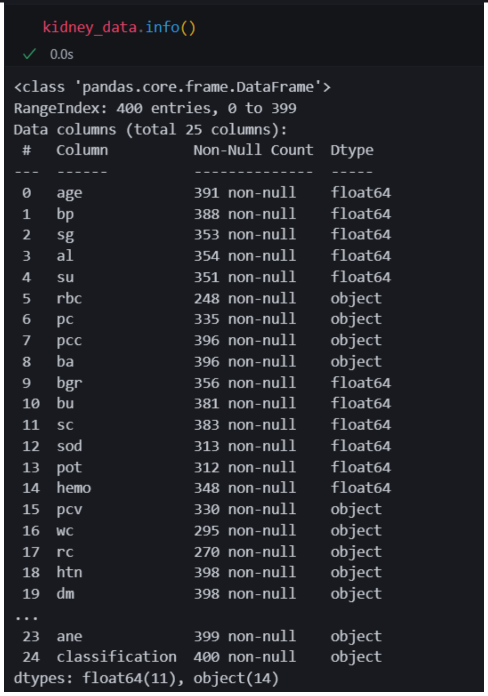
Data Collection and Preprocessing Phase

Date	1th July 2024
Team ID	SWTID1720090815
Project Title	Early Prediction Of Chronic Kidney Disease Using Machine Learning
Maximum Marks	6 Marks

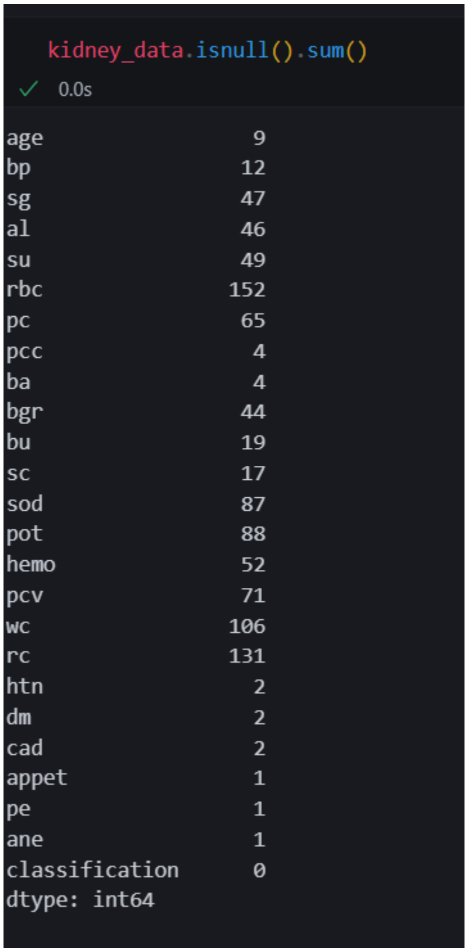
Data Exploration and Preprocessing Template

Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

Section	**Description**
Data Overview	 

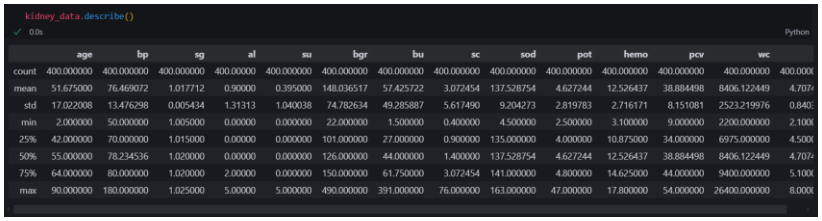
```
kidney_data= pd.read_csv("chronickidneydisease.csv")
kidney_data.reset_index(drop=True,inplace=True)
kidney_data.head()
✓ 0.0s

   id   age    bp     sg   al   su     rbc        pc         pcc          ba  ...  pcv    wc   rc  htn   dm  cad  appet   pe  ane  classification
0   0  48.0  80.0  1.020  1.0  0.0     NaN    normal  notpresent  notpresent  ...   44  7800  5.2  yes  yes   no   good   no   no             ckd
1   1   7.0  50.0  1.020  4.0  0.0     NaN    normal  notpresent  notpresent  ...   38  6000  NaN   no   no   no   good   no   no             ckd
2   2  62.0  80.0  1.010  2.0  3.0  normal    normal  notpresent  notpresent  ...   31  7500  NaN   no  yes   no   poor   no  yes             ckd
3   3  48.0  70.0  1.005  4.0  0.0  normal  abnormal     present  notpresent  ...   32  6700  3.9  yes   no   no   poor  yes  yes             ckd
4   4  51.0  80.0  1.010  2.0  0.0  normal    normal  notpresent  notpresent  ...   35  7300  4.6   no   no   no   good   no   no             ckd
```

```
kidney_data.info()
✓ 0.0s

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 25 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             391 non-null    float64
 1   bp              388 non-null    float64
 2   sg              353 non-null    float64
 3   al              354 non-null    float64
 4   su              351 non-null    float64
 5   rbc             248 non-null    object
 6   pc              335 non-null    object
 7   pcc             396 non-null    object
 8   ba              396 non-null    object
 9   bgr             356 non-null    float64
 10  bu              381 non-null    float64
 11  sc              383 non-null    float64
 12  sod             313 non-null    float64
 13  pot             312 non-null    float64
 14  hemo            348 non-null    float64
 15  pcv             330 non-null    object
 16  wc              295 non-null    object
 17  rc              270 non-null    object
 18  htn             398 non-null    object
 19  dm              398 non-null    object
...
 23  ane             399 non-null    object
 24  classification  400 non-null    object
dtypes: float64(11), object(14)
```



```
kidney_data.isnull().sum()
✓ 0.0s

age                9
bp                12
sg                47
al                46
su                49
rbc              152
pc                65
pcc                4
ba                 4
bgr               44
bu                19
sc                17
sod               87
pot               88
hemo              52
pcv               71
wc               106
rc               131
htn                2
dm                 2
cad                2
appet              1
pe                 1
ane                1
classification     0
dtype: int64
```

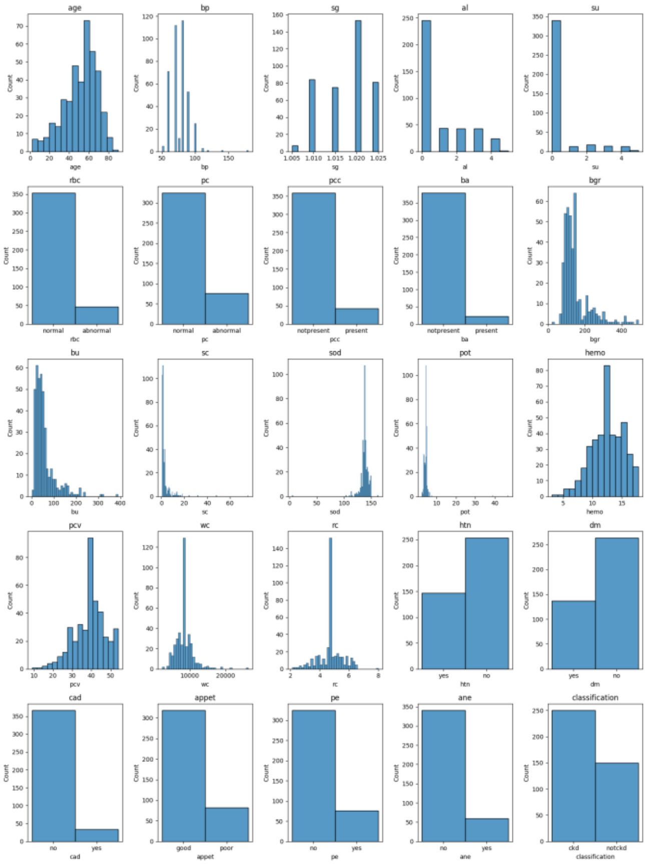
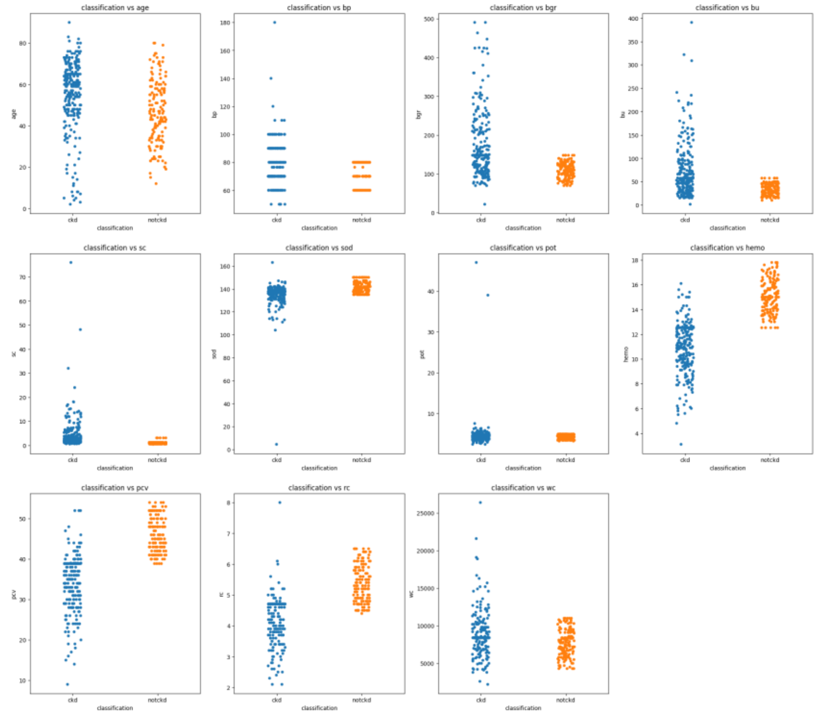
Univariate Analysis
(Pandas describe function,
Histogram)

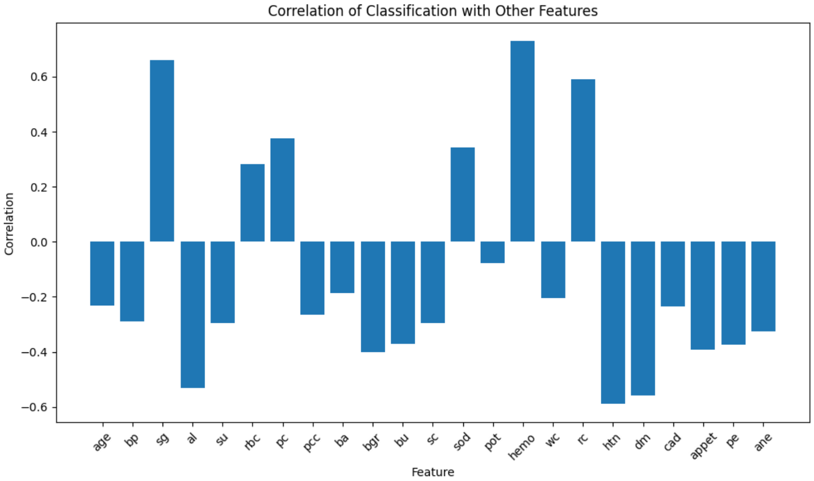
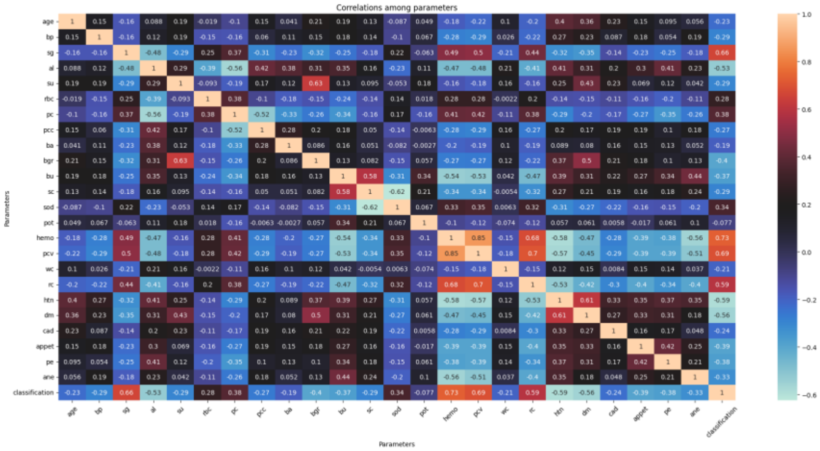


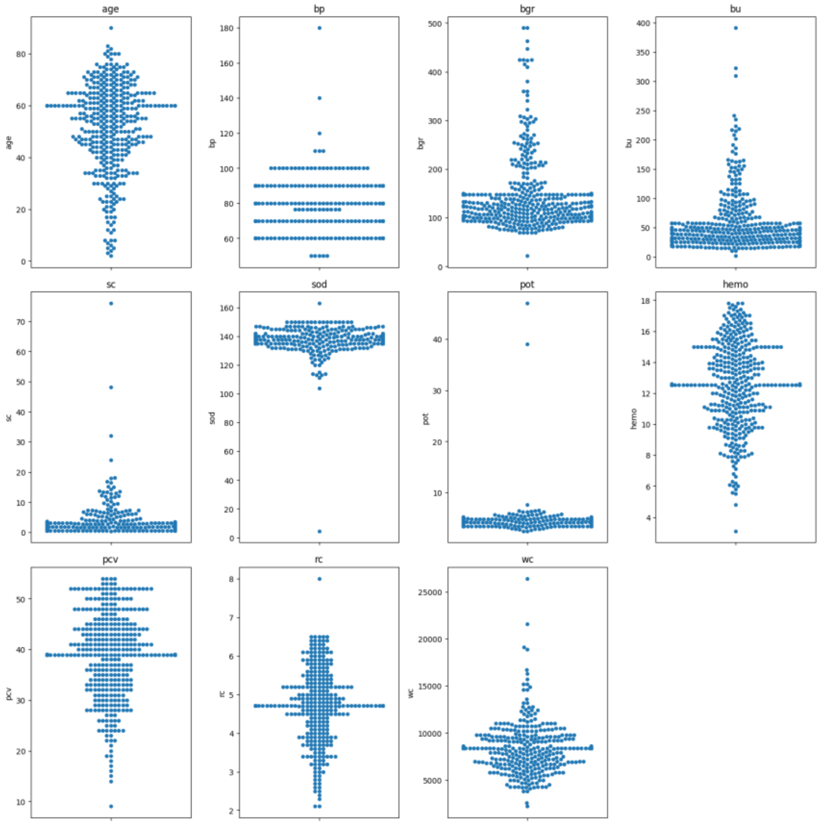
```
kidney_data.describe()
✓ 0.0s
```

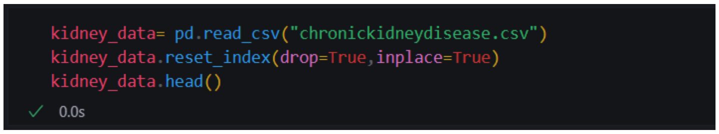
Python

	age	bp	sg	al	su	bgr	bu	sc	sod	pot	hemo	pcv	wc	
count	400.000000	400.000000	400.000000	400.00000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.0000
mean	51.675000	76.469072	1.017712	0.90000	0.395000	148.036517	57.425722	3.072454	137.528754	4.627244	12.526437	38.884498	8406.122449	4.7074
std	17.022008	13.476298	0.005434	1.31313	1.040038	74.782634	49.285887	5.617490	9.204273	2.819783	2.716171	8.151081	2523.219976	0.8403
min	2.000000	50.000000	1.005000	0.00000	0.000000	22.000000	1.500000	0.400000	4.500000	2.500000	3.100000	9.000000	2200.000000	2.1000
25%	42.000000	70.000000	1.015000	0.00000	0.000000	101.000000	27.000000	0.900000	135.000000	4.000000	10.875000	34.000000	6975.000000	4.5000
50%	55.000000	78.234536	1.020000	0.00000	0.000000	126.000000	44.000000	1.400000	137.528754	4.627244	12.526437	38.884498	8406.122449	4.7074
75%	64.000000	80.000000	1.020000	2.00000	0.000000	150.000000	61.750000	3.072454	141.000000	4.800000	14.625000	44.000000	9400.000000	5.1000
max	90.000000	180.000000	1.025000	5.00000	5.000000	490.000000	391.000000	76.000000	163.000000	47.000000	17.800000	54.000000	26400.000000	8.0000

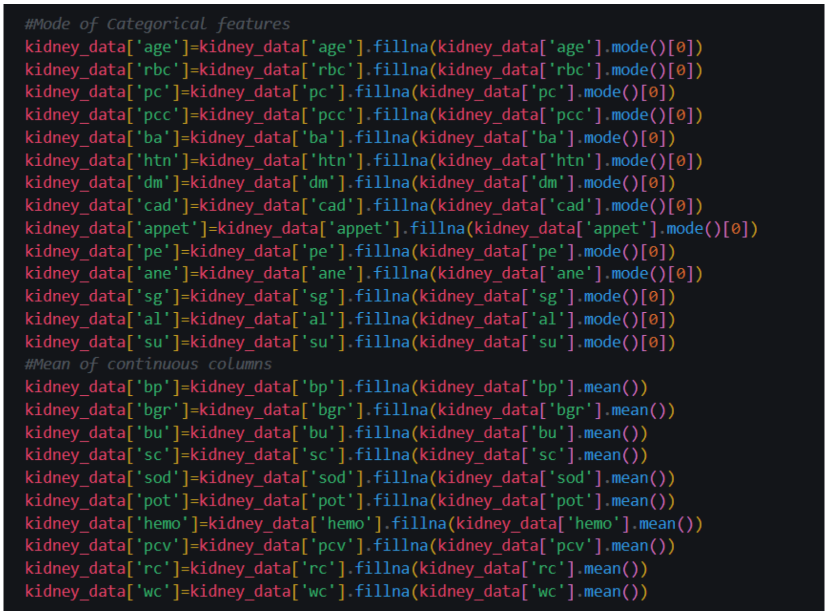
	
Bivariate Analysis (Stripplot, histogram)	

	
Multivariate Analysis (Heatmap)	

<table>
<tr>
<td>Outliers and Anomalies (Swarmplot)</td>
<td>

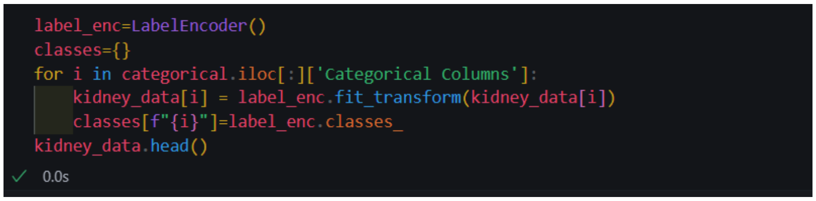
Outliers are purposely retained as in the medical field, there are always a chance of a rare/anomalous case. Thus, retaining outliers may be beneficial for early prediction.</td>
</tr>
<tr>
<td colspan="2"><b>Data Preprocessing Code Screenshots</b></td>
</tr>
<tr>
<td>Loading Data</td>
<td>
<pre>
kidney_data= pd.read_csv("chronickidneydisease.csv")
kidney_data.reset_index(drop=True,inplace=True)
kidney_data.head()
✓ 0.0s
</pre></td>
</tr>
</table>

<table>
<tr>
<td>Handling Missing Data</td>
<td>



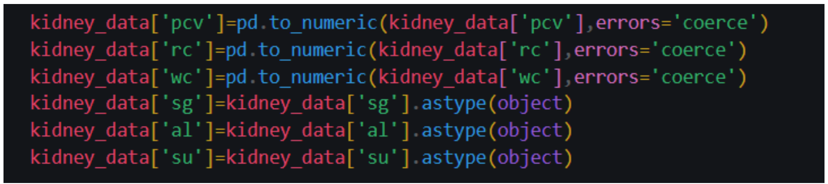
```
#Mode of Categorical features
kidney_data['age']=kidney_data['age'].fillna(kidney_data['age'].mode()[0])
kidney_data['rbc']=kidney_data['rbc'].fillna(kidney_data['rbc'].mode()[0])
kidney_data['pc']=kidney_data['pc'].fillna(kidney_data['pc'].mode()[0])
kidney_data['pcc']=kidney_data['pcc'].fillna(kidney_data['pcc'].mode()[0])
kidney_data['ba']=kidney_data['ba'].fillna(kidney_data['ba'].mode()[0])
kidney_data['htn']=kidney_data['htn'].fillna(kidney_data['htn'].mode()[0])
kidney_data['dm']=kidney_data['dm'].fillna(kidney_data['dm'].mode()[0])
kidney_data['cad']=kidney_data['cad'].fillna(kidney_data['cad'].mode()[0])
kidney_data['appet']=kidney_data['appet'].fillna(kidney_data['appet'].mode()[0])
kidney_data['pe']=kidney_data['pe'].fillna(kidney_data['pe'].mode()[0])
kidney_data['ane']=kidney_data['ane'].fillna(kidney_data['ane'].mode()[0])
kidney_data['sg']=kidney_data['sg'].fillna(kidney_data['sg'].mode()[0])
kidney_data['al']=kidney_data['al'].fillna(kidney_data['al'].mode()[0])
kidney_data['su']=kidney_data['su'].fillna(kidney_data['su'].mode()[0])
#Mean of continuous columns
kidney_data['bp']=kidney_data['bp'].fillna(kidney_data['bp'].mean())
kidney_data['bgr']=kidney_data['bgr'].fillna(kidney_data['bgr'].mean())
kidney_data['bu']=kidney_data['bu'].fillna(kidney_data['bu'].mean())
kidney_data['sc']=kidney_data['sc'].fillna(kidney_data['sc'].mean())
kidney_data['sod']=kidney_data['sod'].fillna(kidney_data['sod'].mean())
kidney_data['pot']=kidney_data['pot'].fillna(kidney_data['pot'].mean())
kidney_data['hemo']=kidney_data['hemo'].fillna(kidney_data['hemo'].mean())
kidney_data['pcv']=kidney_data['pcv'].fillna(kidney_data['pcv'].mean())
kidney_data['rc']=kidney_data['rc'].fillna(kidney_data['rc'].mean())
kidney_data['wc']=kidney_data['wc'].fillna(kidney_data['wc'].mean())
```

</td>
</tr>
<tr>
<td>Data Transformation</td>
<td>

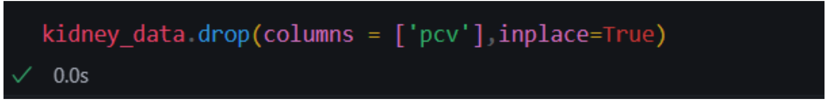


```
label_enc=LabelEncoder()
classes={}
for i in categorical.iloc[:]['Categorical Columns']:
    kidney_data[i] = label_enc.fit_transform(kidney_data[i])
    classes[f"{i}"]=label_enc.classes_
kidney_data.head()
✓ 0.0s
```

</td>
</tr>
<tr>
<td>Feature Engineering</td>
<td>

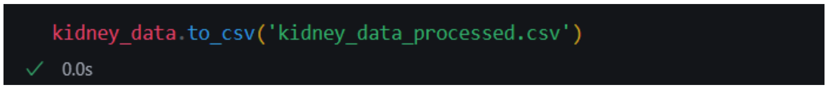


```
kidney_data['pcv']=pd.to_numeric(kidney_data['pcv'],errors='coerce')
kidney_data['rc']=pd.to_numeric(kidney_data['rc'],errors='coerce')
kidney_data['wc']=pd.to_numeric(kidney_data['wc'],errors='coerce')
kidney_data['sg']=kidney_data['sg'].astype(object)
kidney_data['al']=kidney_data['al'].astype(object)
kidney_data['su']=kidney_data['su'].astype(object)
```



```
kidney_data.drop(columns = ['pcv'],inplace=True)
✓ 0.0s
```

</td>
</tr>
<tr>
<td>Save Processed Data</td>
<td>



```
kidney_data.to_csv('kidney_data_processed.csv')
✓ 0.0s
```

</td>
</tr>
</table>