

## ISE CERTIFICATION COURSE DETAILS

NAME:	PRATHVI P SHETTY	USN:	4AL16IS037
SEMESTER:	8	MENTOR:	Mrs . JAISHMA KUMARI
COURSE NAME:	LEARN ANDROID APPLICATION DEVELOPMENT	DATE:	27/05/2020

### SCREENSHOT:

The screenshot displays the Udemy course interface for 'Learn Android Application Development'. The top navigation bar includes the Udemy logo, course title, progress indicator, and share options. The main content area is divided into two sections: 'Course content' on the right and a code editor on the left. The 'Course content' section lists several topics, including 'Section 6: Customizations' and 'Section 7: Saving Data Locally on the Phone'. The code editor on the left shows the implementation of a RecyclerView adapter, with the following code visible:

```

public class ViewHolder extends RecyclerView.ViewHolder {
    TextView textView;
    public ViewHolder(@NonNull View itemView) {
        super(itemView);
    }
}

@Override
public void onBindViewHolder(@NonNull RecyclerView.ViewHolder viewHolder, int i) {
    // TODO: Implement the logic for binding data to the ViewHolder
}

@Override
public int getItemCount() {
    return 0;
}
    
```

The bottom of the screenshot shows the Windows taskbar with various application icons and the system clock indicating 4:50 PM on 5/26/2020.

### BRIEF REPORT: (POINT-WISE)

- 1). An editable text view that shows completion suggestions automatically while the user is typing. The list of suggestions is displayed in a drop down menu from which the user can choose an item to replace the content of the edit box with.
- 2). When you need to create some custom and reuse the views when it is not provided by the Android Ecosystem. Custom Views can be used as widgets like TextView, EditText etc. In this blog we will talk about the aspect of how can we create our own Custom View.
- 3). A toast provides simple feedback about an operation in a small popup. It only fills the amount of space required for the message and the current activity remains visible and interactive. Toasts automatically disappear after a timeout