



T.C.
MANİSA CELAL BAYAR UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEER DEPARTMENT



TO DO LIST (Web Service Supported)

MOBILE COMPUTING WITH IOS AND ANDROID FINAL PROJECT

PREPARED BY

160316054 – CAN ÇALIŞKAN

160315053 – AYŞE BETÜL ERTAN

160315058 – FİRUZE DAMLA ERYILMAZ

İçindekiler

What is To Do List?	3
Application Properties.....	3
Application View	4
Code Part	6
ayarlar.java	6
changeInformationRequest.java	9
changePassword.java	10
changePasswordRequest.java	12
DatePickerFragment.java	13
itemClickListener.java	13
KayitAdapter.java	14
LoginRequest.java	16
main.java	17
RegisterRequest.java	18
signup.java.....	21
TimePickerFragment.java	23
yapilacakEkle.java.....	24
yapilacakEkleRequest.java	32
yapilacakListele.java	32
yapilacaklarListeleRequest.java.....	35
yapilacaklarListeleActivity.java.....	35
yapilacaklarListesiClass.java	38
yapilacajUpdateRequest.java	41
login.php	42
register.php	43
Password.php	43
changeInformation.php.....	53
changePassword.php	54
yapilacakEkle.php	55
yapilacakGuncelle.php	56
yapilacaklarListesi.php	57
References	58

What is To Do List?

Time management is a most important thing our life in this era. We can only remember three or four things at a time. We can easily forget what we will do that day or next day. And this era we can not imagine live without technology or smart phones.

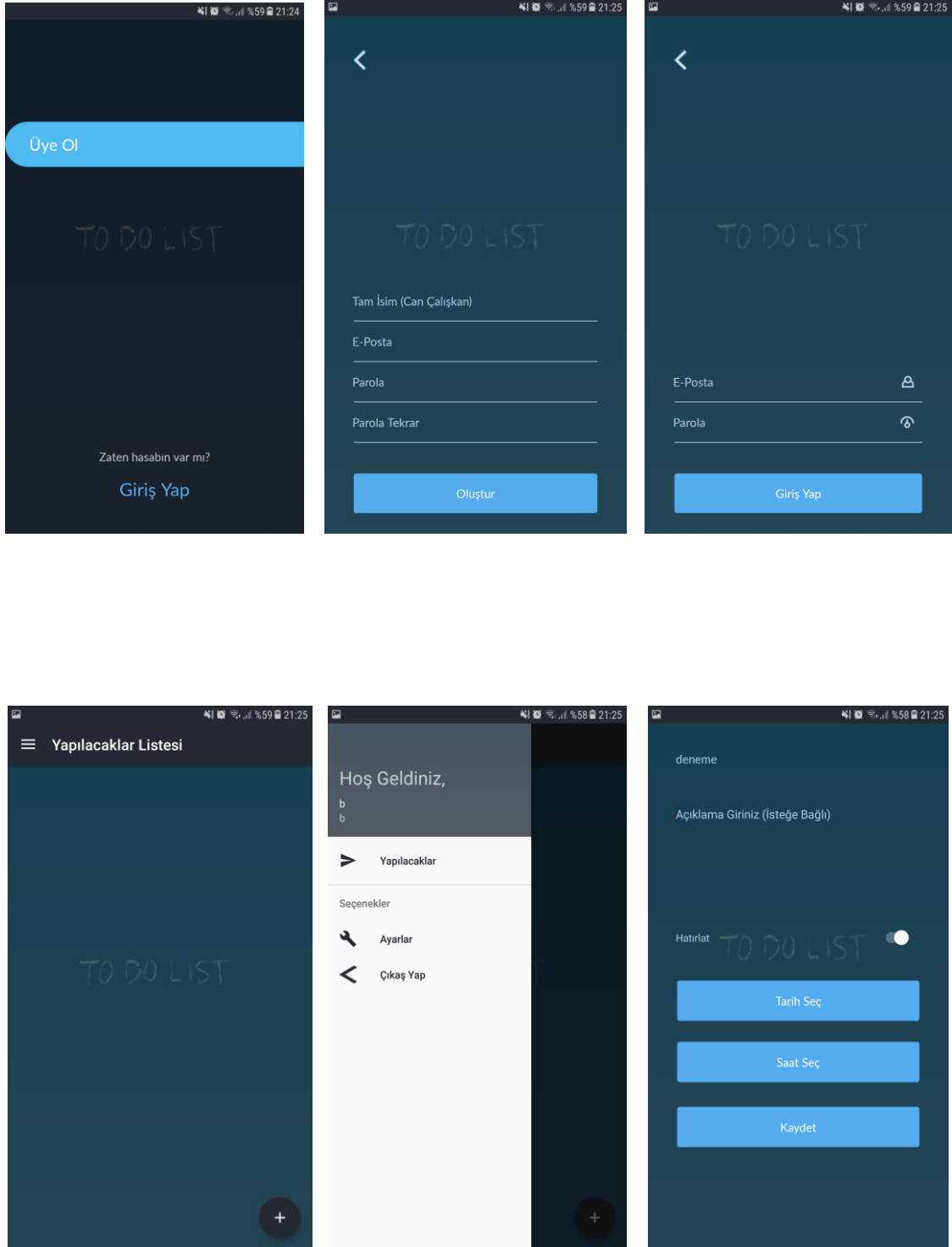
With this requirements of well timing life, we create To Do List application. Our application can solve your forget problems. Also it always near you. You can check, add or remove your daily works or important meeting even your grocery list. And voila! Your life is easier than day before.

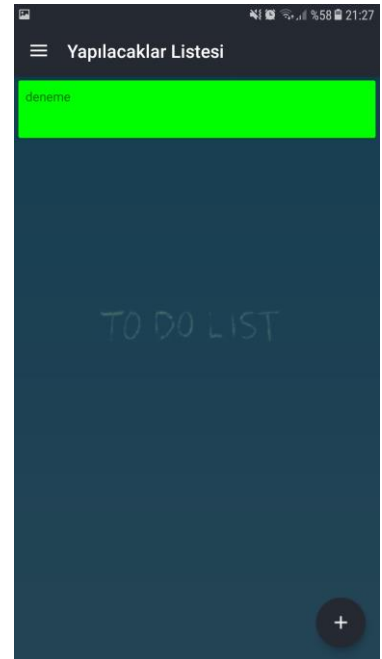
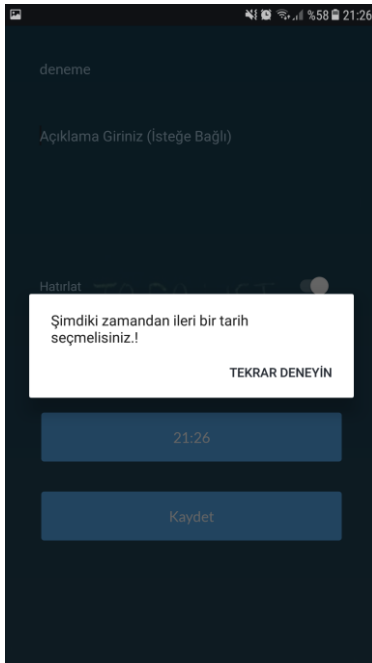
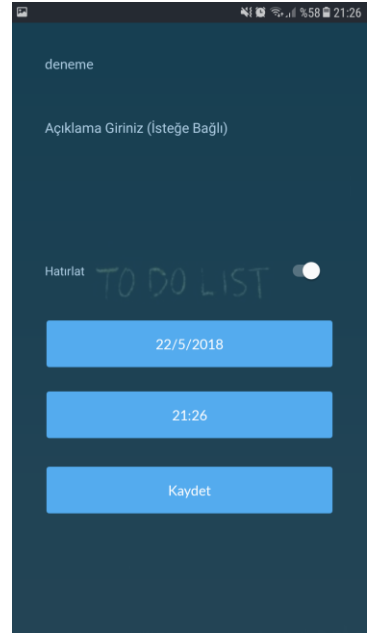
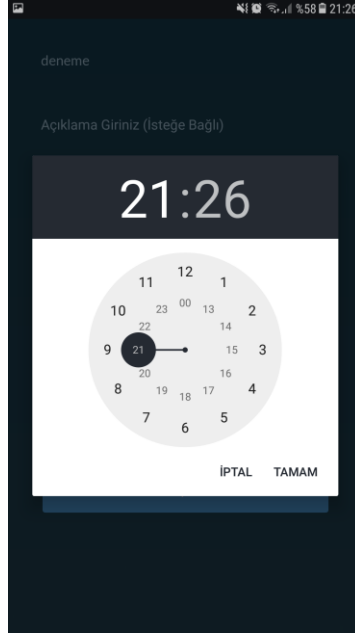
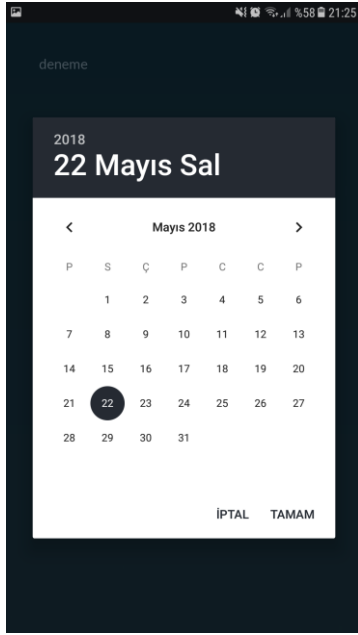
To Do List application is completely work with internet. And every data is taken and stored on database. Users can sign in to application.

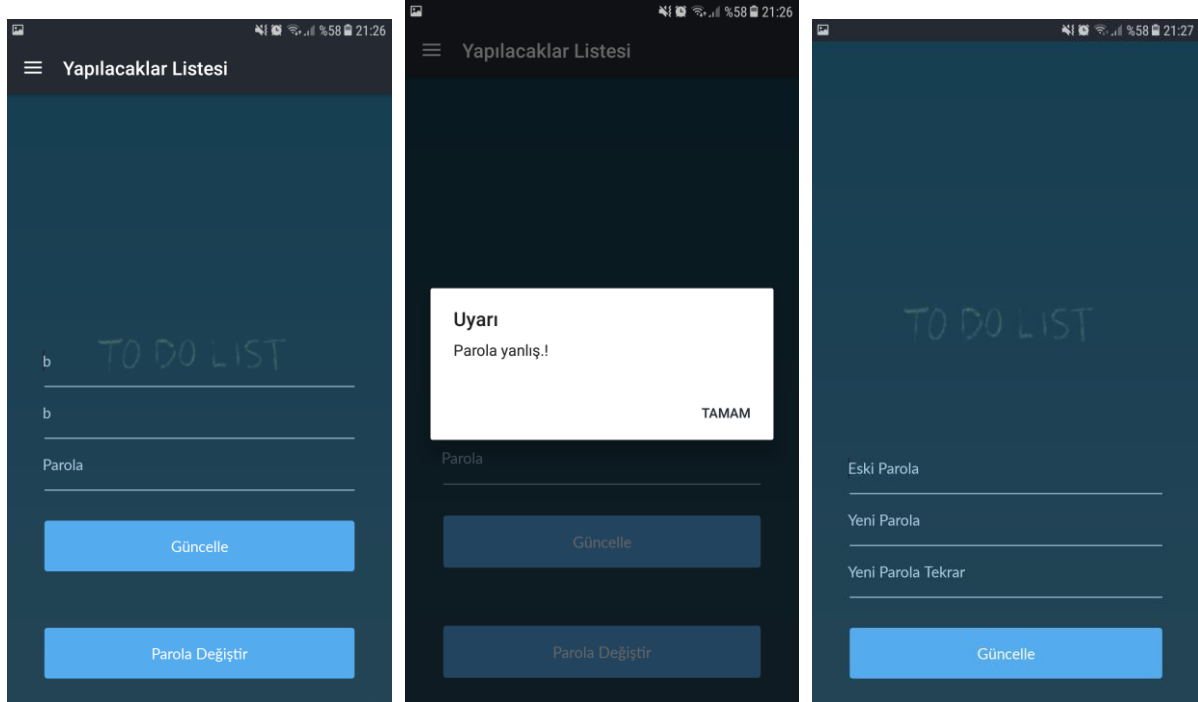
Application Properties

- You can use the application on any android device with internet.
- You must be a member in order to use the application.
- All your records are stored in the database on the server.
- You can use the application synchronously on multiple devices.
- The account is encrypted with the encryption hash algorithm.
- You can set a reminder for a future time.
- If you select the reminder option, the background will be green when the recording time passes.
- You can update the information for a recording you added.
- You can update your account information.
- Once you have logged into the application, your username and password will be remembered.
- It is necessary to enter a title for registration.
- Descriptions can be entered for recording, depending on the requirement.
- Application is supported by web service.

Application View







Code Part

ayarlar.java

```
package com.example.kamran.bluewhite;

import android.app.AlertDialog;
import android.app.Fragment;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.design.widget.FloatingActionButton;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.EditText;
import android.widget.FrameLayout;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.toolbox.Volley;

import org.json.JSONException;
import org.json.JSONObject;
```

```

import static android.content.Context.MODE_PRIVATE;

/**
 * Created by Us on 27.03.2018.
 */

public class Ayarlar extends Fragment {

    private String ISIM_KEY="com.example.kamran.bluewhite.ISIM";
    private String EPOSTA_KEY="com.example.kamran.bluewhite.EPOSTA";
    private String MAIN_KEY="com.example.kamran.bluewhite.MAIN_DATA";
    private String PAROLA_KEY = "com.example.kamran.bluewhite.PAROLA";

    EditText isim, ePosta, parola;
    SharedPreferences sharedPreferences;
    String isimVerisi, epostaVerisi, parolaVerisi;
    TextView parolaGuncelle, guncelleButon;

    @Nullable
    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup
container, Bundle savedInstanceState) {
        final View ViewAyarlar
=inflater.inflate(R.layout.ayarlar_layout, container, false);

        isim = (EditText) ViewAyarlar.findViewById(R.id.fname);
        ePosta = (EditText) ViewAyarlar.findViewById(R.id.mail);
        parola = (EditText) ViewAyarlar.findViewById(R.id.password);

        guncelleButon = (TextView) ViewAyarlar.findViewById(R.id.guncelle);
        parolaGuncelle = (TextView)
ViewAyarlar.findViewById(R.id.parolaDegistir);

        isimVerisi=this.getActivity().getSharedPreferences(MAIN_KEY,
MODE_PRIVATE).getString(ISIM_KEY, "isim bulunamadı");
        epostaVerisi=this.getActivity().getSharedPreferences(MAIN_KEY,
MODE_PRIVATE).getString(EPOSTA_KEY, "eposta bulunamadı");
        parolaVerisi = this.getActivity().getSharedPreferences(MAIN_KEY,
MODE_PRIVATE).getString(PAROLA_KEY, "parola bulunamadı");

        isim.setText(isimVerisi);
        ePosta.setText(epostaVerisi);

        final AlertDialog.Builder dlgAlert = new
AlertDialog.Builder(this.getActivity());

        parolaGuncelle.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                Intent it = new Intent(getActivity(), changePassword.class);
                startActivity(it);

            }
        });

        guncelleButon.setOnClickListener(new View.OnClickListener() {
            @Override

```

```

        public void onClick(View v) {

            if
(parola.getText().toString().equals(parolaVerisi.toString())) {
                Response.Listener<String> responseListener = new
Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {
                        try {
                            JSONObject jsonResponse = new
JSONObject(response);
                            boolean success =
jsonResponse.getBoolean("success");

                            if (success) {
                                SharedPreferences sharedPreferences =
getActivity().getSharedPreferences(MAIN_KEY, MODE_PRIVATE);
                                SharedPreferences.Editor editor =
sharedPreferences.edit();
                                editor.putString(ISIM_KEY,
isim.getText().toString());
                                editor.putString(EPOSTA_KEY,
ePosta.getText().toString());
                                editor.commit();

                                Toast.makeText(getActivity(), "Bilgiler
Güncellendi..", Toast.LENGTH_SHORT).show();

                                Intent intent = new
Intent(getActivity(), yapilacaklarListesiActivity.class);
                                getActivity().startActivity(intent);
                            } else {
                                AlertDialog.Builder message = new
AlertDialog.Builder(getActivity());

                                message.setMessage("İşlem Başarısız.!!")
                                    .setNegativeButton("Tekrar
Deneyin", null)

                                    .create()
                                    .show();
                            }
                        } catch (JSONException e) {
                            e.printStackTrace();
                        }
                    }
                };

                changeInformationRequest changeInformationRequest = new
changeInformationRequest(epostaVerisi.toString(),
isim.getText().toString(), ePosta.getText().toString(), responseListener);
                RequestQueue queue =
Volley.newRequestQueue(getActivity());
                queue.add(changeInformationRequest);
            } else {
                dlgAlert.setMessage("Parola yanlış.!!");
                dlgAlert.setTitle("Uyarı");
                dlgAlert.setPositiveButton("Tamam", null);
                dlgAlert.setCancelable(true);
                dlgAlert.create().show();

                dlgAlert.setPositiveButton("Tamam",

```



```

        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog,
int which) {

                }
            });
        }

    }
});

return ViewAyarlar;
}

}

```

changeInformationRequest.java

```

package com.example.kamran.bluewhite;

import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.toolbox.StringRequest;

import java.util.HashMap;
import java.util.Map;

/**
 * Created by Us on 13.04.2018.
 */

public class changeInformationRequest extends StringRequest {
    private static final String CHANGE_INFORMATION_REQUEST_URL =
"http://todolist.cancaliskan.tk/changeInformation.php";
    private Map<String, String> params;

    public changeInformationRequest(String eski_ePosta, String isim, String
eposta, Response.Listener<String> listener) {
        super(Request.Method.POST, CHANGE_INFORMATION_REQUEST_URL,
listener, null);
        params = new HashMap<>();
        params.put("eski_ePosta", eski_ePosta);
        params.put("isim", isim);
        params.put("eposta", eposta);
    }

    @Override
    public Map<String, String> getParams() {
        return params;
    }
}

```

changePassword.java

```
package com.example.kamran.bluewhite;

import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.toolbox.Volley;

import org.json.JSONException;
import org.json.JSONObject;
import org.w3c.dom.Text;

public class changePassword extends AppCompatActivity {

    private String ISIM_KEY="com.example.kamran.bluewhite.ISIM";
    private String PAROLA_KEY="com.example.kamran.bluewhite.PAROLA";
    private String EPOSTA_KEY="com.example.kamran.bluewhite.EPOSTA";
    private String MAIN_KEY="com.example.kamran.bluewhite.MAIN_DATA";

    EditText eskiParola, yeniParola,yeniParolaTekrar;
    TextView parolaGuncelle;

    String epostaVerisi,parolaVerisi;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.parola_degistir_layout);

        eskiParola = (EditText) findViewById(R.id.eskiParola);
        yeniParola = (EditText) findViewById(R.id.yeniParola);
        yeniParolaTekrar = (EditText) findViewById(R.id.yeniParolaTekrar);
        parolaGuncelle = (TextView) findViewById(R.id.guncelle);

        final AlertDialog.Builder dlgAlert = new
AlertDialog.Builder(changePassword.this);

        parolaGuncelle.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                parolaVerisi=getSharedPreferences(MAIN_KEY,
MODE_PRIVATE).getString(PAROLA_KEY,"eposta bulunamadı");

                if(yeniParola.getText().toString().equals(yeniParolaTekrar.getText().toString()))
                {
                    if (
eskiParola.getText().toString().equals(parolaVerisi.toString())) {
                        Response.Listener<String> responseListener = new
Response.Listener<String>() {
```

```

@Override
public void onResponse(String response) {
    try {
        JSONObject jsonResponse = new
JSONObject(response);

        boolean success =
jsonResponse.getBoolean("success");

        if (success) {
            SharedPreferences sharedPreferences
= getSharedPreferences(MAIN_KEY, MODE_PRIVATE);
            SharedPreferences.Editor
editor=sharedPreferences.edit();
            editor.putString(PAROLA_KEY,
yeniParola.getText().toString());
            editor.commit();

            Toast.makeText(changePassword.this,
"Parola Değiştirildi..", Toast.LENGTH_SHORT).show();

            Intent intent = new
Intent(changePassword.this, yapilacaklarListesiActivity.class);
            changePassword.this.startActivity(intent);
        } else {
            AlertDialog.Builder message = new
AlertDialog.Builder(changePassword.this);

            message.setMessage("İşlem
Başarısız.!!")
                .setNegativeButton("Tekrar
Deneyin", null)
                .create()
                .show();
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }
}

epostaVerisi = getSharedPreferences(MAIN_KEY,
MODE_PRIVATE).getString(EPOSTA_KEY, "eposta bulunamadı");

changePasswordRequest changePasswordRequest = new
changePasswordRequest(epostaVerisi, yeniParola.getText().toString(),
responseListener);

RequestQueue queue =
Volley.newRequestQueue(changePassword.this);
queue.add(changePasswordRequest);
}
else
{
    dlgAlert.setMessage("Eski parolanızı yanlış.!!");
    dlgAlert.setTitle("Uyarı");
    dlgAlert.setPositiveButton("Tamam", null);
    dlgAlert.setCancelable(true);
    dlgAlert.create().show();

    dlgAlert.setPositiveButton("Tamam",

```

```

        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface
dialog, int which) {

                }
            });
        }
    }
    else
    {
        dlgAlert.setMessage("Yeni parola eşleşmiyor!");
        dlgAlert.setTitle("Uyarı");
        dlgAlert.setPositiveButton("Tamam", null);
        dlgAlert.setCancelable(true);
        dlgAlert.create().show();

        dlgAlert.setPositiveButton("Tamam",
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog,
int which) {

                    }
                });
            }
        });
    }
}

```

changePasswordRequest.java

```

package com.example.kamran.bluewhite;

import android.widget.EditText;

import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.toolbox.StringRequest;

import java.util.HashMap;
import java.util.Map;

/**
 * Created by Us on 11.04.2018.
 */

public class changePasswordRequest extends StringRequest {
    private static final String CHANGE_PASSWORD_REQUEST_URL =
"http://todolist.cancaliskan.tk/changePassword.php";
    private Map<String, String> params;

    public changePasswordRequest(String ePosta, String password,
Response.Listener<String> listener) {
        super(Request.Method.POST, CHANGE_PASSWORD_REQUEST_URL, listener,
null);
        params = new HashMap<>();
    }
}

```

```

        params.put("eposta", ePosta);
        params.put("parola", password);
    }

    @Override
    public Map<String, String> getParams() {
        return params;
    }
}

```

DatePickerFragment.java

```

package com.example.kamran.bluewhite;

import android.app.DatePickerDialog;
import android.app.Dialog;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.v4.app.DialogFragment;

import java.util.Calendar;

/**
 * Created by Us on 13.04.2018.
 */

public class DatePickerFragment extends DialogFragment {
    @NonNull
    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        Calendar c = Calendar.getInstance();
        int year = c.get(Calendar.YEAR);
        int month = c.get(Calendar.MONTH);
        int day = c.get(Calendar.DAY_OF_MONTH);
        return new DatePickerDialog(getActivity(),
            (DatePickerDialog.OnDateSetListener) getActivity(), year, month, day);
    }
}

```

ItemClickListener.java

```

package com.example.kamran.bluewhite;

import android.view.View;

/**
 * Created by Us on 15.04.2018.
 */

public interface ItemClickListener {
    void onItemClick(int pos);
}

```

KayitAdapter.java

```
package com.example.kamran.bluewhite;

import android.content.Context;
import android.content.Intent;
import android.graphics.Color;
import android.support.v7.widget.CardView;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;
import java.util.List;

/**
 * Created by Us on 15.04.2018.
 */

public class KayitAdapter extends
RecyclerView.Adapter<KayitAdapter.KayitViewHolder> {

    private Context ctx;
    private ArrayList<yapilacaklarListesiClass> kayitList;

    public KayitAdapter(Context ctx, ArrayList<yapilacaklarListesiClass>
kayitList) {
        this.ctx = ctx;
        this.kayitList = kayitList;
    }

    @Override
    public KayitViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {
        LayoutInflater inflater = LayoutInflater.from(ctx);
        View view = inflater.inflate(R.layout.list_yapilacaklar, null);
        return new KayitViewHolder(view);
    }

    public yapilacaklarListesiClass kayitDetay;

    @Override
    public void onBindViewHolder(KayitViewHolder holder, int position) {
        yapilacaklarListesiClass yapilacaklarListesiClass =
kayitList.get(position);
        holder.baslik.setText(yapilacaklarListesiClass.getBaslik());
        holder.aciklama.setText(yapilacaklarListesiClass.getMetin());

        if (yapilacaklarListesiClass.getHatirlat().equals("Evet")) {
            if (Calendar.getInstance().get(Calendar.YEAR) >=
yapilacaklarListesiClass.getYil(yapilacaklarListesiClass.getTarih()))
            {

```

```

        if(Calendar.getInstance().get(Calendar.YEAR) >
yapilacaklarListesiClass.getYil(yapilacaklarListesiClass.getTarih()))
        {
            holder.cardView.setCardBackgroundColor(Color.GREEN);
        }
        else {
            if ((Calendar.getInstance().get(Calendar.MONTH)+1) >=
yapilacaklarListesiClass.getAy(yapilacaklarListesiClass.getTarih())) {
                if((Calendar.getInstance().get(Calendar.MONTH)+1) >
yapilacaklarListesiClass.getAy(yapilacaklarListesiClass.getTarih()))
                {

holder.cardView.setCardBackgroundColor(Color.GREEN);
                }
                else {
                    if
                    (Calendar.getInstance().get(Calendar.DAY_OF_MONTH) >=
yapilacaklarListesiClass.getGun(yapilacaklarListesiClass.getTarih())) {

if(Calendar.getInstance().get(Calendar.DAY_OF_MONTH) >
yapilacaklarListesiClass.getGun(yapilacaklarListesiClass.getTarih()))
                    {

holder.cardView.setCardBackgroundColor(Color.GREEN);
                    }
                    else {
                        if
                        (Calendar.getInstance().get(Calendar.HOUR_OF_DAY) >=
yapilacaklarListesiClass.getSaatBolumu(yapilacaklarListesiClass.getSaat()))
                        {

if(Calendar.getInstance().get(Calendar.HOUR_OF_DAY) >
yapilacaklarListesiClass.getSaatBolumu(yapilacaklarListesiClass.getSaat()))
                            {

holder.cardView.setCardBackgroundColor(Color.GREEN);
                            }
                            else {
                                if
                                (Calendar.getInstance().get(Calendar.MINUTE) >=
yapilacaklarListesiClass.get DakikaBolumu(yapilacaklarListesiClass.getSaat()
)) {

holder.cardView.setCardBackgroundColor(Color.GREEN);
                                }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
holder.setItemClickListener(new itemClickListener() {
    @Override
    public void onItemClick(int pos) {
        kayitDetay = kayitList.get(pos);
        openDetailActivity(kayitDetay);
    }
});
});

```

```

    }

    private void openDetailActivity(yapilacaklarListesiClass kayitDetay) {
        Intent intent = new Intent(ctx, yapilacakEkle.class);
        intent.putExtra("ID", (kayitDetay.getID() + ""));
        intent.putExtra("eposta", kayitDetay.getEposta());
        intent.putExtra("baslik", kayitDetay.getBaslik());
        intent.putExtra("metin", kayitDetay.getMetin());
        intent.putExtra("hatirlat", kayitDetay.getHatirlat());
        intent.putExtra("tarih", kayitDetay.getTarih());
        intent.putExtra("saat", kayitDetay.getSaat());
        ctx.startActivity(intent);
    }

    @Override
    public int getItemCount() {
        return kayitList.size();
    }

    class KayitViewHolder extends RecyclerView.ViewHolder implements
    View.OnClickListener {

        itemClickListener itemClickListener;
        TextView baslik, aciklama;
        CardView cardView;

        public KayitViewHolder(View itemView) {
            super(itemView);

            int yil, ay, gun, saat, dakika;
            cardView = (CardView) itemView.findViewById(R.id.cardView);
            baslik = (TextView) itemView.findViewById(R.id.baslik);
            aciklama = (TextView) itemView.findViewById(R.id.aciklama);

            itemView.setOnClickListener(this);
        }

        @Override
        public void onClick(View v) {
            this.itemClickListener.onItemClick(this.getLayoutPosition());
        }

        public void setItemClickListener(itemClickListener
itemClickListener) {
            this.itemClickListener = itemClickListener;
        }
    }
}

```

LoginRequest.java

```

package com.example.kamran.bluewhite;

import com.android.volley.Response;

```



```

import com.android.volley.toolbox.StringRequest;

import java.util.HashMap;
import java.util.Map;

/**
 * Created by Us on 26.03.2018.
 */

public class LoginRequest extends StringRequest {
    private static final String
LOGIN_REQUEST_URL="http://todolist.cancaliskan.tk/login.php";
    private Map<String, String> params;

    public LoginRequest(String eposta, String parola,
Response.Listener<String> listener)
    {
        super(Method.POST, LOGIN_REQUEST_URL, listener, null);
        params = new HashMap<>();
        params.put("eposta", eposta);
        params.put("parola", parola);
    }

    @Override
    public Map<String, String> getParams() {
        return params;
    }
}

```

main.java

```

package com.example.kamran.bluewhite;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.LinearLayout;
import android.widget.TextView;

public class main extends AppCompatActivity {
    TextView sin;
    LinearLayout circle;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        circle = (LinearLayout)findViewById(R.id.circle);
        sin = (TextView)findViewById(R.id.sin);

        circle.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

```

```

        Intent it = new Intent(main.this, signup.class);
        startActivity(it);

    }

});
sin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent it = new Intent(main.this, signin.class);
        startActivity(it);
    }
});
}
}

```

RegisterRequest.java

```

package com.example.kamran.bluewhite;

/**
 * Created by Us on 25.03.2018.
 */
import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.android.volley.toolbox.StringRequest;
import com.android.volley.toolbox.Volley;
import java.util.HashMap;
import java.util.Map;

public class RegisterRequest extends StringRequest {
    private static final String
REGISTER_REQUEST_URL="http://todolist.cancaliskan.tk/register.php";
    private Map<String, String> params;

    public RegisterRequest(String isim, String eposta, String parola,
Response.Listener<String> listener)
    {
        super(Method.POST, REGISTER_REQUEST_URL, listener, null);
        params = new HashMap<>();
        params.put("isim", isim);
        params.put("eposta", eposta);
        params.put("parola", parola);
    }

    @Override
    public Map<String, String> getParams(){
        return params;
    }
}

```

signin.java

```

package com.example.kamran.bluewhite;

import android.app.AlertDialog;
import android.content.DialogInterface;

```

```

import android.content.Intent;
import android.content.SharedPreferences;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.toolbox.Volley;

import org.json.JSONException;
import org.json.JSONObject;

public class signin extends AppCompatActivity {

    private String ISIM_KEY="com.example.kamran.bluewhite.ISIM";
    private String EPOSTA_KEY="com.example.kamran.bluewhite.EPOSTA";
    private String PAROLA_KEY="com.example.kamran.bluewhite.PAROLA";
    private String MAIN_KEY="com.example.kamran.bluewhite.MAIN_DATA";

    SharedPreferences sharedPreferences;
    SharedPreferences.Editor editor;

    EditText mail,password;
    TextView sin;
    ImageView sback;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_signin);
        sback = (ImageView) findViewById(R.id.sinb);
        sback.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent it = new Intent(signin.this,main.class);
                startActivity(it);
            }
        });

        sin = (TextView) findViewById(R.id.sin);
        mail = (EditText) findViewById(R.id.usrusr);
        password = (EditText) findViewById(R.id.pswrd);

        final AlertDialog.Builder dlgAlert = new
        AlertDialog.Builder(signin.this);

        sin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                final String mailText = mail.getText().toString();
                final String passwordText = password.getText().toString();

                if(mailText.equals("") || passwordText.equals(""))
                {
                    dlgAlert.setMessage("Lütfen tüm alanları
doldurunuz.!");

```

```

        dlgAlert.setTitle("Uyarı");
        dlgAlert.setPositiveButton("Tamam", null);
        dlgAlert.setCancelable(true);
        dlgAlert.create().show();

        dlgAlert.setPositiveButton("Tamam",
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog,
int which) {

                    }
                });
        }
        else
        {
            Response.Listener<String> responseListener = new
Response.Listener<String>() {
                @Override
                public void onResponse(String response) {
                    try {
                        JSONObject jsonResponse = new
JSONObject(response);

                        boolean success =
jsonResponse.getBoolean("success");

                        if(success){
                            String isim =
jsonResponse.getString("isim");
                            String eposta =
jsonResponse.getString("eposta");
                            String parola =
jsonResponse.getString("parola");

                            sharedPreferences=getSharedPreferences(MAIN_KEY, MODE_PRIVATE);
                            editor=sharedPreferences.edit();
                            editor.putString(ISIM_KEY, isim);
                            editor.putString(EPOSTA_KEY, eposta);
                            editor.putString(PAROLA_KEY, parola);
                            editor.commit();
                            Toast.makeText(signin.this, "Giriş
Başarılı", Toast.LENGTH_SHORT).show();

                            Intent intent = new Intent(signin.this,
yapilacaklarListesiActivity.class);
                            intent.putExtra("isim", isim);
                            intent.putExtra("eposta", eposta);
                            signin.this.startActivity(intent);
                        }
                        else
                        {
                            AlertDialog.Builder message = new
AlertDialog.Builder(signin.this);

                            message.setMessage("İşlem Başarısız!")
                                .setNegativeButton("Tekrar
Deneyin", null)

                                .create()
                                .show();
                        }
                    }
                }
            }
        }
    }
}

```

```

        }
        catch (JSONException e) {
            e.printStackTrace();
        }
    }
};

LoginRequest loginRequest = new
LoginRequest(mailText,passwordText,responseListener);
RequestQueue queue =
Volley.newRequestQueue(signin.this);
queue.add(loginRequest);

    }
}
});
}
}

```

signup.java

```

package com.example.kamran.bluewhite;

import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.ImageView;
import android.widget.*;

import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.toolbox.Volley;

import org.json.JSONException;
import org.json.JSONObject;

public class signup extends AppCompatActivity
{
    EditText fname,mail,password,password2;
    TextView sin;
    ImageView sback;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_signup);
        sback = (ImageView)findViewById(R.id.sback);
        sback.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v)
            {
                Intent it = new Intent(signup.this, main.class);
                startActivity(it);
            }
        });
    }
}

```

```

});
sin = (TextView) findViewById(R.id.sin);
mail = (EditText) findViewById(R.id.mail);
fname = (EditText) findViewById(R.id.fname);
password = (EditText) findViewById(R.id.password);
password2 = (EditText) findViewById(R.id.password2);

sin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v)
    {
        final String isim = fname.getText().toString();
        final String eposta = mail.getText().toString();
        final String parola = password.getText().toString();

        AlertDialog.Builder dlgAlert = new
AlertDialog.Builder(signup.this);

        if (fname.getText().toString().equals("")
            || mail.getText().toString().equals("")
            || password.getText().toString().equals("")
            || password2.getText().toString().equals("")) {

            dlgAlert.setMessage("Lütfen tüm alanları
doldurunuz.!");

            dlgAlert.setTitle("Uyarı");
            dlgAlert.setPositiveButton("Tamam", null);
            dlgAlert.setCancelable(true);
            dlgAlert.create().show();

            dlgAlert.setPositiveButton("Tamam",
                new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog,
int which) {

                        }
                    });
        }
        else
        {

            if(password.getText().toString().equals(password2.getText().toString()))
            {
                Response.Listener<String> responseListener = new
Response.Listener<String>() {
                    @Override
                    public void onResponse(String response) {
                        try {
                            JSONObject jsonResponse = new
JSONObject(response);

                            boolean success =
jsonResponse.getBoolean("success");

                            if(success){
                                Intent intent = new
Intent(signup.this, signin.class);

                                signup.this.startActivity(intent);
                            }
                            else
                            {

```



```

/**
 * Created by Us on 13.04.2018.
 */

public class TimePickerFragment extends DialogFragment {

    @NonNull
    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        Calendar c = Calendar.getInstance();
        int hour = c.get(Calendar.HOUR_OF_DAY);
        int minute = c.get(Calendar.MINUTE);

        return new TimePickerDialog(getActivity(),
            (TimePickerDialog.OnTimeSetListener) getActivity(), hour, minute,
            DateFormat.is24HourFormat(getActivity()));
    }
}

```

yapilacakEkle.java

```

package com.example.kamran.bluewhite;

import android.app.AlertDialog;
import android.app.DatePickerDialog;
import android.app.TimePickerDialog;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.graphics.Color;
import android.support.v4.app.DialogFragment;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CompoundButton;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.Switch;
import android.widget.TextView;
import android.widget.TimePicker;
import android.widget.Toast;

import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.toolbox.Volley;

import org.json.JSONException;
import org.json.JSONObject;

import java.text.DateFormat;
import java.util.Calendar;

public class yapilacakEkle extends AppCompatActivity implements
    DatePickerDialog.OnDateSetListener, TimePickerDialog.OnTimeSetListener {

    private String ISIM_KEY = "com.example.kamran.bluewhite.ISIM";
    private String EPOSTA_KEY = "com.example.kamran.bluewhite.EPOSTA";
    private String MAIN_KEY = "com.example.kamran.bluewhite.MAIN_DATA";
    private String PAROLA_KEY = "com.example.kamran.bluewhite.PAROLA";
}

```



```

EditText baslik, metin;
TextView tarihSec, saatSec, yapilacakEkle;
Switch hatirlat;

String epostaVerisi;
public String
intent_ID,intent_eposta,intent_baslik,intent_metin,intent_hatirlat,intent_t
arih,intent_saat;

public int yil,ay,gun,saat,dakika;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_yapilacak_ekle);

    tarihSec = (TextView) findViewById(R.id.tarihSecButon);
    saatSec = (TextView) findViewById(R.id.saatSecButon);
    hatirlat = (Switch) findViewById(R.id.yapilacakHatirlat);
    yapilacakEkle = (TextView) findViewById(R.id.yapilacakEkleButon);

    baslik = (EditText) findViewById(R.id.yapilacakBaslik);
    metin = (EditText) findViewById(R.id.yapilacakAciklama);

    Intent intent = this.getIntent();
    Bundle extras = intent.getExtras();
    if(extras != null) {
        intent_ID = intent.getExtras().getString("ID");
        intent_eposta = intent.getExtras().getString("eposta");
        intent_baslik = intent.getExtras().getString("baslik");
        intent_metin = intent.getExtras().getString("metin");
        intent_hatirlat = intent.getExtras().getString("hatirlat");
        intent_tarih = intent.getExtras().getString("tarih");
        intent_saat = intent.getExtras().getString("saat");
    }
    if (intent_ID != null) {
        baslik.setText(intent_baslik);
        metin.setText(intent_metin);
        if (intent_hatirlat.equals("Evet")) {
            hatirlat.setChecked(true);
        } else {
            hatirlat.setChecked(false);
        }
        if (!intent_tarih.equals(""))
            tarihSec.setText(intent_tarih);
        else
            tarihSec.setText("Tarih Seç");
        if (!intent_saat.equals(""))
            saatSec.setText(intent_saat);
        else
            saatSec.setText("Saat Seç");
    }

    if (!hatirlat.isChecked()) {
        tarihSec.setVisibility(View.GONE);
        saatSec.setVisibility(View.GONE);
    } else {
        tarihSec.setVisibility(View.VISIBLE);
        saatSec.setVisibility(View.VISIBLE);
    }
}

```

```

        hatirlat.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
        if (!hatirlat.isChecked()) {
            tarihSec.setVisibility(View.GONE);
            saatSec.setVisibility(View.GONE);
        } else {
            tarihSec.setVisibility(View.VISIBLE);
            saatSec.setVisibility(View.VISIBLE);
        }
    }
});

tarihSec.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        DialogFragment datePicker = new DatePickerFragment();
        datePicker.show(getSupportFragmentManager(), "date
picker");
    }
});

saatSec.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        DialogFragment timePicker = new TimePickerFragment();
        timePicker.show(getSupportFragmentManager(), "time
picker");
    }
});

yapilacakEkle.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (intent_ID != null) {
            /// güncelleme işlemi başlangıcı ///
            if (!baslik.getText().toString().equals("")) {
                if (hatirlat.isChecked()
                    &&
(!tarihSec.getText().toString().equals("Tarih Seç")
                    &&
!saatSec.getText().toString().equals("Saat Seç"))) {
                    if (Calendar.getInstance().get(Calendar.YEAR)
<= yil)
                        {
                            if(Calendar.getInstance().get(Calendar.YEAR) < yil)
                                {
                                    kayitGuncelleHatirlatmali();
                                }
                            else {
                                if
((Calendar.getInstance().get(Calendar.MONTH)) <= ay) {
                                    if((Calendar.getInstance().get(Calendar.MONTH)) < ay)
                                        {
                                            kayitGuncelleHatirlatmali();
                                        }
                                    else {

```



```

Toast.LENGTH_SHORT).show();

Intent intent = new
Intent(yapilacakEkle.this, yapilacaklarListesiActivity.class);

yapilacakEkle.this.startActivity(intent);
} else {
    alertDialog("İşlem
Başarısız.!");
}
} catch (JSONException e) {
    e.printStackTrace();
}
};
yapilacakUpdateRequest yapilacakUpdateRequest =
new yapilacakUpdateRequest(intent_ID, baslik.getText().toString(),
metin.getText().toString(),
"Hayır", "", "", responseListener);
RequestQueue queue =
Volley.newRequestQueue(yapilacakEkle.this);
queue.add(yapilacakUpdateRequest);
}
} else {
    alertDialog("Başlık boş geçilemez.!");
}
}
/// güncelleme işlemi bitti ///
else {
    /// ekleme işlemi başladı ///
    if (!baslik.getText().toString().equals("")) {
        if (hatirlat.isChecked()
            &&
(!tarihSec.getText().toString().equals("Tarih Seç")
            &&
!saatSec.getText().toString().equals("Saat Seç"))) {
            if (Calendar.getInstance().get(Calendar.YEAR)
<= yil)
            {

if(Calendar.getInstance().get(Calendar.YEAR) < yil)
            {
                kayitEklemeHatirlatmali();
            }
            else {
                if
((Calendar.getInstance().get(Calendar.MONTH)) <= ay) {
if((Calendar.getInstance().get(Calendar.MONTH)) < ay)
                {
                    kayitEklemeHatirlatmali();
                }
                else {
                    if
(Calendar.getInstance().get(Calendar.DAY_OF_MONTH) <= gun) {
if(Calendar.getInstance().get(Calendar.DAY_OF_MONTH) < gun)
                    {

kayitEklemeHatirlatmali();

                    }

```



```

        c.set(Calendar.DAY_OF_MONTH, dayOfMonth);
        String currentDateString =
DateFormat.getDateInstance(DateFormat.DATE_FIELD).format(c.getTime());
        tarihSec.setText(dayOfMonth+"/"+(monthOfYear+1)+"/"+year);//ay
        sıfırdan başladığı için +1
    }

    @Override
    public void onTimeSet(TimePicker view, int hourOfDay, int minute) {

        saat=hourOfDay;
        dakika=minute;

        saatSec.setText(hourOfDay + ":" + minute);
    }

    public void alertDialog(String mesaj)
    {
        AlertDialog.Builder message = new
        AlertDialog.Builder(yapilacakEkle.this);

        message.setMessage(mesaj)
            .setNegativeButton("Tekrar Deneyin", null)
            .create()
            .show();
    }

    public void kayitGuncelleHatirlatmali()
    {
        Response.Listener<String> responseListener = new
        Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                try {
                    JSONObject jsonResponse = new JSONObject(response);
                    boolean success = jsonResponse.getBoolean("success");
                    if (success) {
                        Toast.makeText(yapilacakEkle.this, "İşlem
Başarılı", Toast.LENGTH_SHORT).show();
                        Intent intent = new Intent(yapilacakEkle.this,
yapilacaklarListesiActivity.class);
                        yapilacakEkle.this.startActivity(intent);
                    } else {
                        alertDialog("İşlem Başarısız!");
                    }
                } catch (JSONException e) {
                    e.printStackTrace();
                }
            }
        };
        yapilacakUpdateRequest yapilacakUpdateRequest = new
        yapilacakUpdateRequest(intent_ID, baslik.getText().toString(),
        metin.getText().toString(),
        "Evet", tarihSec.getText().toString(),
        saatSec.getText().toString(), responseListener);
        RequestQueue queue = Volley.newRequestQueue(yapilacakEkle.this);
        queue.add(yapilacakUpdateRequest);
    }

    public void kayitEklemeHatirlatmali()
    {

```

```

        Response.Listener<String> responseListener = new
Response.Listener<String>() {
    @Override
    public void onResponse(String response) {
        try {
            JSONObject jsonResponse = new JSONObject(response);
            boolean success = jsonResponse.getBoolean("success");

            if (success) {
                Toast.makeText(yapilacakEkle.this, "İşlem
Başarılı", Toast.LENGTH_SHORT).show();

                Intent intent = new Intent(yapilacakEkle.this,
yapilacaklarListesiActivity.class);
                yapilacakEkle.this.startActivity(intent);
            } else {
                alertDialog("İşlem Başarısız!");
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
};

epostaVerisi=getSharedPreferences(MAIN_KEY,
MODE_PRIVATE).getString(EPOSTA_KEY,"eposta bulunamadı");
yapilacakEkleRequest yapilacakEkleRequest = new
yapilacakEkleRequest(epostaVerisi, baslik.getText().toString(),
metin.getText().toString(),
    "Evet", tarihSec.getText().toString(),
saatSec.getText().toString(), responseListener);
RequestQueue queue = Volley.newRequestQueue(yapilacakEkle.this);
queue.add(yapilacakEkleRequest);
}

public void kayitEklemeHatirlatmasiz()
{
    Response.Listener<String> responseListener = new
Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            try {
                JSONObject jsonResponse = new JSONObject(response);
                boolean success = jsonResponse.getBoolean("success");

                if (success) {
                    Toast.makeText(yapilacakEkle.this, "İşlem
Başarılı", Toast.LENGTH_SHORT).show();

                    Intent intent = new Intent(yapilacakEkle.this,
yapilacaklarListesiActivity.class);
                    yapilacakEkle.this.startActivity(intent);
                } else {
                    alertDialog("İşlem Başarısız!");
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    };

    epostaVerisi=getSharedPreferences(MAIN_KEY,
MODE_PRIVATE).getString(EPOSTA_KEY,"eposta bulunamadı");

```

```

        yapılacakEkleRequest yapılacakEkleRequest = new
        yapılacakEkleRequest(epostaVerisi, baslik.getText().toString(),
        metin.getText().toString(),
            "Hayır", "", "", responseListener);
        RequestQueue queue = Volley.newRequestQueue(yapilacakEkle.this);
        queue.add(yapilacakEkleRequest);
    }
}

```

yapilacakEkleRequest.java

```

package com.example.kamran.bluewhite;

import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.toolbox.StringRequest;

import java.util.Date;
import java.util.HashMap;
import java.util.Map;

/**
 * Created by Us on 14.04.2018.
 */

public class yapılacakEkleRequest extends StringRequest {
    private static final String YAPILACAK_EKLE_REQUEST_URL =
        "http://todolist.cancaliskan.tk/yapilacakEkle.php";
    private Map<String, String> params;

    public yapılacakEkleRequest(String ePosta, String baslik,
        String metin, String hatirlat, String
        tarih, String saat, Response.Listener<String> listener) {
        super(Request.Method.POST, YAPILACAK_EKLE_REQUEST_URL, listener,
        null);
        params = new HashMap<>();
        params.put("eposta", ePosta);
        params.put("baslik", baslik);
        params.put("metin", metin);
        params.put("hatirlat", hatirlat);
        params.put("tarih", tarih);
        params.put("saat", saat);
    }

    @Override
    public Map<String, String> getParams() {
        return params;
    }
}

```

yapilacakListele.java

```

package com.example.kamran.bluewhite;

import android.app.AlertDialog;

```



```

import android.app.Fragment;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.RequestQueue;
import com.android.volley.Response;
import com.android.volley.toolbox.Volley;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.util.ArrayList;

import static android.content.Context.MODE_PRIVATE;

/**
 * Created by Us on 27.03.2018.
 */

public class yapilacaklarListele extends Fragment {
    private String ISIM_KEY="com.example.kamran.bluewhite.ISIM";
    private String EPOSTA_KEY="com.example.kamran.bluewhite.EPOSTA";
    private String MAIN_KEY="com.example.kamran.bluewhite.MAIN_DATA";
    private String PAROLA_KEY = "com.example.kamran.bluewhite.PAROLA";

    String epostaVerisi;

    SharedPreferences sharedPreferences;

    RecyclerView recyclerView;
    KayitAdapter[] adapter = new KayitAdapter[1];
    ArrayList<yapilacaklarListesiClass> yapilacalarListesiArray = new
    ArrayList<yapilacaklarListesiClass>();

    @Nullable
    @Override
    public View onCreateView(LayoutInflater inflater, @Nullable ViewGroup
    container, Bundle savedInstanceState) {
        final View yapilacaklarListele
    =inflater.inflate(R.layout.activity_yapilacaklar,container,false);

        recyclerView = (RecyclerView)
    yapilacaklarListele.findViewById(R.id.recycler_view);
        recyclerView.setHasFixedSize(true);
        recyclerView.setLayoutManager(new

```

```

LinearLayoutManager(getActivity()));

        yapılacaklarListesi();

        return yapılacaklarListele;
    }

    private void yapılacaklarListesi()
    {
        Response.Listener<String> responseListener = new
Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                try {
                    ProgressDialog progressDialog = new
ProgressDialog(getActivity());
                    progressDialog.setMessage("Kayıtlar yükleniyor..");
                    progressDialog.show();
                    JSONObject jsonResponse = new JSONObject(response);

                    JSONArray array =
jsonResponse.getJSONArray("yapilacaklarListesi");
                    for (int i=0; i< array.length(); i++)
                    {
                        JSONObject jo = array.getJSONObject(i);
                        yapılacaklarListesiClass yapılacakKayit = new
yapilacaklarListesiClass(jo.getInt("ID"),
jo.getString("eposta"),jo.getString("baslik")
,jo.getString("metin"),jo.getString("hatirlat"),jo.getString("tarih"),jo.ge
tString("saat"));

                        yapilacalarListesiArray.add(yapilacakKayit);
                    }
                    progressDialog.dismiss();

                    adapter[0] = new KayitAdapter(getActivity(),
yapilacalarListesiArray);
                    recyclerView.setAdapter(adapter[0]);
                }
                catch (JSONException e) {
                    e.printStackTrace();
                }
            }
        };

        epostaVerisi=this.getActivity().getSharedPreferences(MAIN_KEY,
MODE_PRIVATE).getString(EPOSTA_KEY,"eposta bulunamadı");

        yapılacaklarListeleRequest yapılacaklarListeleRequest = new
yapilacaklarListeleRequest(epostaVerisi,responseListener);
        RequestQueue queue = Volley.newRequestQueue(getActivity());
        queue.add(yapilacaklarListeleRequest);
    }

}

```

yapilacaklarListeleRequest.java

```
package com.example.kamran.bluewhite;

import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.toolbox.StringRequest;

import java.util.HashMap;
import java.util.Map;

/**
 * Created by Us on 14.04.2018.
 */

public class yapilacaklarListeleRequest extends StringRequest {
    private static final String YAPILACAK_Listele_REQUEST_URL =
        "http://todolist.cancaliskan.tk/yapilacaklarListesi.php";
    private Map<String, String> params;

    public yapilacaklarListeleRequest(String ePosta,
        Response.Listener<String> listener) {
        super(Request.Method.POST, YAPILACAK_Listele_REQUEST_URL, listener,
            null);
        params = new HashMap<>();
        params.put("eposta", ePosta);
    }

    @Override
    public Map<String, String> getParams() {
        return params;
    }
}
```

yapilacaklarListeleActivity.java

```
package com.example.kamran.bluewhite;

import android.app.FragmentManager;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.view.View;
import android.support.design.widget.NavigationView;
import android.support.v4.view.GravityCompat;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBarDrawerToggle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.TextView;
import android.widget.Toast;
```

```

public class yapilacaklarListesiActivity extends AppCompatActivity
    implements NavigationView.OnNavigationItemSelectedListener {

    private String ISIM_KEY = "com.example.kamran.bluewhite.ISIM";
    private String EPOSTA_KEY = "com.example.kamran.bluewhite.EPOSTA";
    private String MAIN_KEY = "com.example.kamran.bluewhite.MAIN_DATA";

    SharedPreferences sharedPreferences;

    String isimVerisi, epostaVerisi;

    TextView navBarIsim, navBarEposta;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_yapilacaklar_listesi);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        FloatingActionButton fab = (FloatingActionButton)
findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                /* Snackbar.make(view, "Replace with your own action",
Snackbar.LENGTH_LONG)
                    .setAction("Action", null).show();*/
                Intent intent = new Intent(yapilacaklarListesiActivity.this,
yapilacakEkle.class);
                yapilacaklarListesiActivity.this.startActivity(intent);
            }
        });

        setTitle("Yapilacaklar Listesi");

        if (getSharedPreferences(MAIN_KEY,
MODE_PRIVATE).getString(EPOSTA_KEY, "eposta bulunamadı").equals("eposta
bulunamadı")) {
            Intent intent = new Intent(yapilacaklarListesiActivity.this,
main.class);
            yapilacaklarListesiActivity.this.startActivity(intent);
        }

        DrawerLayout drawer = (DrawerLayout)
findViewById(R.id.drawer_layout);
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
            this, drawer, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close);
        drawer.addDrawerListener(toggle);
        toggle.syncState();

        NavigationView navigationView = (NavigationView)
findViewById(R.id.nav_view);
        navigationView.setNavigationItemSelectedListener(this);
    }

    @Override
    public void onBackPressed() {

```

```

        DrawerLayout drawer = (DrawerLayout)
findViewById(R.id.drawer_layout);
        if (drawer.isDrawerOpen(GravityCompat.START)) {
            drawer.closeDrawer(GravityCompat.START);
        } else {
            super.onBackPressed();
        }
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is
        present.
        // getMenuInflater().inflate(R.menu.yapilacaklar_listesi, menu);
        navBarIsim = (TextView) findViewById(R.id.navBarIsim);
        navBarEposta = (TextView) findViewById(R.id.navBarEposta);

        isimVerisi = getSharedPreferences(MAIN_KEY,
MODE_PRIVATE).getString(ISIM_KEY, "isim bulunamadı");
        epostaVerisi = getSharedPreferences(MAIN_KEY,
MODE_PRIVATE).getString(EPOSTA_KEY, "eposta bulunamadı");

        navBarIsim.setText(isimVerisi);
        navBarEposta.setText(epostaVerisi);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        /*if (id == R.id.action_settings) {
            return true;
        }*/

        return super.onOptionsItemSelected(item);
    }

    @SuppressWarnings("StatementWithEmptyBody")
    @Override
    public boolean onNavigationItemSelected(MenuItem item) {
        // Handle navigation view item clicks here.
        int id = item.getItemId();

        FragmentManager fragmentManager = getFragmentManager();

        /*if (id == R.id.nav_camera) {
            // Handle the camera action
        } else if (id == R.id.nav_gallery) {

        } else if (id == R.id.nav_slideshow) {

        } else if (id == R.id.nav_manage) {

        } else if (id == R.id.nav_share) {

        } else if (id == R.id.nav_send) {

```

```

        }*/
        if (id == R.id.nav_cikisYap) {
            sharedPreferences = getSharedPreferences(MAIN KEY,
MODE_PRIVATE);
            sharedPreferences.edit().clear().commit();
            Toast.makeText(yapilacaklarListesiActivity.this, "Çıkış
Başarılı", Toast.LENGTH_SHORT).show();

            Intent intent = new Intent(yapilacaklarListesiActivity.this,
yapilacaklarListesiActivity.class);
            yapilacaklarListesiActivity.this.startActivity(intent);
        } else if (id == R.id.nav_yapilacaklar) {
            FloatingActionButton fab = (FloatingActionButton)
findViewById(R.id.fab);
            fab.show();

            fragmentManager.beginTransaction().replace(R.id.content_FrameLayout, new
yapilacaklarListele()).commit();
        } else if (id == R.id.nav_ayarlar) {
            FloatingActionButton fab = (FloatingActionButton)
findViewById(R.id.fab);
            fab.hide();

            fragmentManager.beginTransaction().replace(R.id.content_FrameLayout, new
Ayarlar()).commit();
        }

        DrawerLayout drawer = (DrawerLayout)
findViewById(R.id.drawer_layout);
        drawer.closeDrawer(GravityCompat.START);
        return true;
    }
}

```

yapilacaklarListesiClass.java

```

package com.example.kamran.bluewhite;

/**
 * Created by Us on 14.04.2018.
 */

public class yapilacaklarListesiClass {
    private String eposta,baslik,metin,hatirlat,tarih,saat;
    private int ID;
    public yapilacaklarListesiClass(int ID, String eposta, String baslik,
String metin, String hatirlat, String tarih, String saat ) {
        this.ID=ID;
        this.eposta=eposta;
        this.baslik=baslik;
        this.metin=metin;
        this.hatirlat=hatirlat;
        this.tarih=tarih;
        this.saat=saat;
    }

    public String getEposta() {
        return eposta;
    }
}

```

```

    }

    public void setEposta(String eposta) {
        this.eposta = eposta;
    }

    public String getBaslik() {
        return baslik;
    }

    public void setBaslik(String baslik) {
        this.baslik = baslik;
    }

    public String getMetin() {
        return metin;
    }

    public void setMetin(String metin) {
        this.metin = metin;
    }

    public String getHatirlat() {
        return hatirlat;
    }

    public void setHatirlat(String hatirlat) {
        this.hatirlat = hatirlat;
    }

    public String getTarih() {
        return tarih;
    }

    public void setTarih(String tarih) {
        this.tarih = tarih;
    }

    public String getSaat() {
        return saat;
    }

    public void setSaat(String saat) {
        this.saat = saat;
    }

    public int getID() {
        return ID;
    }

    public void setID(int ID) {
        this.ID = ID;
    }

    public int getGun(String tarih)
    {
        String tempGun="";
        for (int i = 0; i < 3; i++)
        {
            if(tarih.charAt(i) != '/')
            {

```

```

        tempGun += tarih.charAt(i);
    }
    else
        break;
}
return Integer.parseInt(tempGun);
}

public int getAy(String tarih)
{
    String slashCheck="";
    String tempAy="";
    for (int i = 0; i < tarih.length(); i++)
    {
        if(slashCheck.equals("/") && tarih.charAt(i) != '/')
        {
            tempAy += tarih.charAt(i);
        }
        if(tarih.charAt(i) == '/')
        {
            slashCheck += "/";
        }
    }
    return Integer.parseInt(tempAy);
}

public int getYil(String tarih)
{
    String slashCheck="";
    String tempYil="";
    for (int i = 0; i < tarih.length(); i++)
    {
        if(slashCheck.equals("//"))
        {
            tempYil += tarih.charAt(i);
        }
        if(tarih.charAt(i) == '/')
        {
            slashCheck += "/";
        }
    }
    return Integer.parseInt(tempYil);
}

public int getSaatBolumu(String saat) {
    String slashCheck="";
    String tempSaat="";

    for (int i = 0; i < saat.length(); i++)
    {
        if(saat.charAt(i) != ':')
        {
            tempSaat += saat.charAt(i);
        }
        else
            break;
    }
    return Integer.parseInt(tempSaat);
}

public int getDakikaBolumu(String saat) {

```



```

String slashCheck="";
String tempDakika="";

for (int i = 0; i < saat.length(); i++)
{
    if(slashCheck.equals(":"))
    {
        tempDakika += saat.charAt(i);
    }
    if(saat.charAt(i) == ':')
    {
        slashCheck += ":";
    }
}
return Integer.parseInt(tempDakika);
}
}

```

yapilacajUpdateRequest.java

```

package com.example.kamran.bluewhite;

import com.android.volley.Request;
import com.android.volley.Response;
import com.android.volley.toolbox.StringRequest;

import java.util.HashMap;
import java.util.Map;

/**
 * Created by Us on 15.04.2018.
 */

public class yapilacakUpdateRequest extends StringRequest {
    private static final String YAPILACAK_Update_REQUEST_URL =
"http://todolist.cancaliskan.tk/yapilacakGuncelle.php";
    private Map<String, String> params;

    public yapilacakUpdateRequest(String ID, String baslik,
String metin, String hatirlat, String
tarih,String saat, Response.Listener<String> listener) {
        super(Request.Method.POST, YAPILACAK_Update_REQUEST_URL, listener,
null);
        params = new HashMap<>();
        params.put("ID", ID);
        params.put("baslik", baslik);
        params.put("metin", metin);
        params.put("hatirlat", hatirlat);
        params.put("tarih", tarih);
        params.put("saat", saat);
    }

    @Override
    public Map<String, String> getParams() {
        return params;
    }
}

```

login.php

```
<?php
```

```
require("password.php");
```

```
$con = mysqli_connect("localhost:3306", "cancaliskan", "Ee6!r3m7", "cancaliskan_toDoList");
```

```
$eposta = $_POST["eposta"];
```

```
$parola = $_POST["parola"];
```

```
$statement = mysqli_prepare($con, "SELECT * FROM user WHERE eposta = ? AND parola = ?");
```

```
mysqli_stmt_bind_param($statement, "ss", $eposta, $parola);
```

```
mysqli_stmt_execute($statement);
```

```
mysqli_stmt_store_result($statement);
```

```
mysqli_stmt_bind_result($statement, $ID, $isim, $eposta, $parola);
```

```
$response = array();
```

```
$response["success"] = false;
```

```
while(mysqli_stmt_fetch($statement)){
```

```
    if (password_verify($parola, $parola)) {
```

```
        $response["success"] = true;
```

```
        $response["isim"] = $isim;
```

```
        $response["eposta"] = $eposta;
```

```
        // $response["parola"] = $parola;
```

```
    }
```

```
}
```

```
echo json_encode($response);
```

```
?>
```

register.php

```
<?php
    $con = mysqli_connect("localhost:3306", "cancaliskan", "Ee6!r3m7", "cancaliskan_toDoList");

    $isim = $_POST["isim"];
    $eposta = $_POST["eposta"];
    $parola = $_POST["parola"];

    $statement = mysqli_prepare($con, "INSERT INTO kullanicilar (isim, eposta, parola) VALUES (?, ?, ?, ?)");

    mysqli_stmt_bind_param($statement, "sss", $isim, $eposta, $parola);
    mysqli_stmt_execute($statement);

    $response = array();
    $response["success"] = true;

    echo json_encode($response);
?>
```

Password.php

```
<?php
/**
 * A Compatibility library with PHP 5.5's simplified password hashing API.
 *
 * @author Anthony Ferrara <ircmaxell@php.net>
 * @license http://www.opensource.org/licenses/mit-license.html MIT License
 * @copyright 2012 The Authors
 */
```

```

namespace {

if (!defined('PASSWORD_BCRYPT')) {

    /**
     * PHPUnit Process isolation caches constants, but not function declarations.
     * So we need to check if the constants are defined separately from
     * the functions to enable supporting process isolation in userland
     * code.
     */
    define('PASSWORD_BCRYPT', 1);
    define('PASSWORD_DEFAULT', PASSWORD_BCRYPT);
    define('PASSWORD_BCRYPT_DEFAULT_COST', 10);
}

if (!function_exists('password_hash')) {

    /**
     * Hash the password using the specified algorithm
     *
     * @param string $password The password to hash
     * @param int $algo The algorithm to use (Defined by PASSWORD_* constants)
     * @param array $options The options for the algorithm to use
     *
     * @return string|false The hashed password, or false on error.
     */
    function password_hash($password, $algo, array $options = array()) {

        if (!function_exists('crypt')) {
            trigger_error("Crypt must be loaded for password_hash to function", E_USER_WARNING);
            return null;
        }

        if (is_null($password) || is_int($password)) {
            $password = (string) $password;
        }

        if (!is_string($password)) {

```

```

        trigger_error("password_hash(): Password must be a string", E_USER_WARNING);

        return null;
    }

    if (!is_int($algo)) {
        trigger_error("password_hash() expects parameter 2 to be long, " . gettype($algo) . " given",
            E_USER_WARNING);
        return null;
    }

    $resultLength = 0;
    switch ($algo) {
        case PASSWORD_BCRYPT:
            $cost = PASSWORD_BCRYPT_DEFAULT_COST;

            if (isset($options['cost'])) {
                $cost = (int) $options['cost'];

                if ($cost < 4 || $cost > 31) {
                    trigger_error(sprintf("password_hash(): Invalid bcrypt cost parameter specified: %d",
                        $cost), E_USER_WARNING);
                    return null;
                }
            }

            // The length of salt to generate
            $raw_salt_len = 16;

            // The length required in the final serialization
            $required_salt_len = 22;

            $hash_format = sprintf("$2y$%02d$", $cost);

            // The expected length of the final crypt() output
            $resultLength = 60;

            break;

        default:
            trigger_error(sprintf("password_hash(): Unknown password hashing algorithm: %s",
                $algo), E_USER_WARNING);
            return null;
    }

```

```

    }

    $salt_req_encoding = false;

    if (isset($options['salt'])) {
        switch (gettype($options['salt'])) {
            case 'NULL':

            case 'boolean':

            case 'integer':

            case 'double':

            case 'string':
                $salt = (string) $options['salt'];

                break;

            case 'object':
                if (method_exists($options['salt'], '__toString')) {
                    $salt = (string) $options['salt'];

                    break;
                }

            case 'array':

            case 'resource':

            default:
                trigger_error('password_hash(): Non-string salt parameter supplied',
E_USER_WARNING);

                return null;
        }

        if (PasswordCompat\binary_strlen($salt) < $required_salt_len) {
            trigger_error(sprintf("password_hash(): Provided salt is too short: %d expecting %d",
PasswordCompat\binary_strlen($salt), $required_salt_len), E_USER_WARNING);

            return null;
        } elseif (0 == preg_match('#^[a-zA-Z0-9./]+${#D}', $salt)) {
            $salt_req_encoding = true;
        }
    } else {
        $buffer = "";
    }

```

```

$buffer_valid = false;

if (function_exists('mcrypt_create_iv') && !defined('PHALANGER')) {
    $buffer = mcrypt_create_iv($raw_salt_len, MCRYPT_DEV_URANDOM);
    if ($buffer) {
        $buffer_valid = true;
    }
}

if (!$buffer_valid && function_exists('openssl_random_pseudo_bytes')) {
    $strong = false;
    $buffer = openssl_random_pseudo_bytes($raw_salt_len, $strong);
    if ($buffer && $strong) {
        $buffer_valid = true;
    }
}

if (!$buffer_valid && @is_readable('/dev/urandom')) {
    $file = fopen('/dev/urandom', 'r');
    $read = 0;
    $local_buffer = "";
    while ($read < $raw_salt_len) {
        $local_buffer .= fread($file, $raw_salt_len - $read);
        $read = PasswordCompat\binary\_strlen($local_buffer);
    }
    fclose($file);
    if ($read >= $raw_salt_len) {
        $buffer_valid = true;
    }
    $buffer = str_pad($buffer, $raw_salt_len, "\0") ^ str_pad($local_buffer, $raw_salt_len,
"\0");
}

if (!$buffer_valid || PasswordCompat\binary\_strlen($buffer) < $raw_salt_len) {
    $buffer_length = PasswordCompat\binary\_strlen($buffer);

```

```

    for ($i = 0; $i < $raw_salt_len; $i++) {
        if ($i < $buffer_length) {
            $buffer[$i] = $buffer[$i] ^ chr(mt_rand(0, 255));
        } else {
            $buffer .= chr(mt_rand(0, 255));
        }
    }
}

$salt = $buffer;

$salt_req_encoding = true;
}

if ($salt_req_encoding) {
    // encode string with the Base64 variant used by crypt
    $base64_digits =
        'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/';
    $bcrypt64_digits =
        './ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789';
    $base64_string = base64_encode($salt);
    $salt = strtr(rtrim($base64_string, '='), $base64_digits, $bcrypt64_digits);
}

$salt = PasswordCompat\binary\_substr($salt, 0, $required_salt_len);

$hash = $hash_format . $salt;

$ret = crypt($password, $hash);

if (!is_string($ret) || PasswordCompat\binary\_strlen($ret) != $resultLength) {
    return false;
}

return $ret;
}

/**
 * Get information about the password hash. Returns an array of the information
 * that was used to generate the password hash.

```



```

*
* array(
*   'algo' => 1,
*   'algoName' => 'bcrypt',
*   'options' => array(
*     'cost' => PASSWORD_BCRYPT_DEFAULT_COST,
*   ),
* )
*
* @param string $hash The password hash to extract info from
*
* @return array The array of information about the hash.
*/
function password_get_info($hash) {
    $return = array(
        'algo' => 0,
        'algoName' => 'unknown',
        'options' => array(),
    );

    if (PasswordCompat\binary\_substr($hash, 0, 4) == '$2y$' &&
        PasswordCompat\binary\_strlen($hash) == 60) {
        $return['algo'] = PASSWORD_BCRYPT;
        $return['algoName'] = 'bcrypt';
        list($cost) = sscanf($hash, "$2y$%d$");
        $return['options']['cost'] = $cost;
    }

    return $return;
}

/**
 * Determine if the password hash needs to be rehashed according to the options provided
 */

```

```

* If the answer is true, after validating the password using password_verify, rehash it.
*
* @param string $hash The hash to test
* @param int $algo The algorithm used for new password hashes
* @param array $options The options array passed to password_hash
*
* @return boolean True if the password needs to be rehashed.
*/
function password_needs_rehash($hash, $algo, array $options = array()) {
    $info = password_get_info($hash);
    if ($info['algo'] !== (int) $algo) {
        return true;
    }
    switch ($algo) {
        case PASSWORD_BCRYPT:
            $cost = isset($options['cost']) ? (int) $options['cost'] :
PASSWORD_BCRYPT_DEFAULT_COST;
            if ($cost !== $info['options']['cost']) {
                return true;
            }
            break;
    }
    return false;
}

/**
* Verify a password against a hash using a timing attack resistant approach
*
* @param string $password The password to verify
* @param string $hash The hash to verify against
*
* @return boolean If the password matches the hash

```

```

*/
function password_verify($password, $hash) {
    if (!function_exists('crypt')) {
        trigger_error("Crypt must be loaded for password_verify to function", E_USER_WARNING);
        return false;
    }

    $ret = crypt($password, $hash);

    if (!is_string($ret) || PasswordCompat\binary\_strlen($ret) !=
    PasswordCompat\binary\_strlen($hash) || PasswordCompat\binary\_strlen($ret) <= 13) {
        return false;
    }

    $status = 0;
    for ($i = 0; $i < PasswordCompat\binary\_strlen($ret); $i++) {
        $status |= (ord($ret[$i]) ^ ord($hash[$i]));
    }

    return $status === 0;
}
}

namespace PasswordCompat\binary {
    if (!function_exists('PasswordCompat\\binary\\_strlen')) {
        /**
         * Count the number of bytes in a string
         *
         * We cannot simply use strlen() for this, because it might be overwritten by the mbstring
         extension.
         *
         * In this case, strlen() will count the number of *characters* based on the internal encoding. A
         * sequence of bytes might be regarded as a single multibyte character.
         *
         * @param string $binary_string The input string
         *
         * @internal

```

```

* @return int The number of bytes
*/
function _strlen($binary_string) {
    if (function_exists('mb_strlen')) {
        return mb_strlen($binary_string, '8bit');
    }
    return strlen($binary_string);
}

/**
 * Get a substring based on byte limits
 *
 * @see _strlen()
 *
 * @param string $binary_string The input string
 * @param int $start
 * @param int $length
 *
 * @internal
 * @return string The substring
 */
function _substr($binary_string, $start, $length) {
    if (function_exists('mb_substr')) {
        return mb_substr($binary_string, $start, $length, '8bit');
    }
    return substr($binary_string, $start, $length);
}

/**
 * Check if current PHP version is compatible with the library
 *
 * @return boolean the check result
 */

```

```

function check() {
    static $pass = NULL;
    if (is_null($pass)) {
        if (function_exists('crypt')) {
            $hash = '$2y$04$usesomesillystringfore7hnbRJHxXVLeakoG8K30oukPsA.ztMG';
            $test = crypt("password", $hash);
            $pass = $test == $hash;
        } else {
            $pass = false;
        }
    }
    return $pass;
}
}
}

```

changeInformation.php

```

<?php
$con = mysqli_connect("localhost:3306", "cancaliskan", "Ee6!r3m7", "cancaliskan_toDoList");

$eski_ePosta = $_POST["eski_ePosta"];
$isim = $_POST["isim"];
$eposta = $_POST["eposta"];

$stmt = mysqli_prepare($con, "SELECT * FROM kullanicilar WHERE eposta = ?");
mysqli_stmt_bind_param($stmt, "s", $eski_ePosta);
mysqli_stmt_execute($stmt);
mysqli_stmt_store_result($stmt);
mysqli_stmt_bind_result($stmt, $colID, $colIsim, $colEposta, $colParola);

$response = array();

```

```

$response["success"] = false;

while(mysqli_stmt_fetch($statement)){
    if ($eski_ePosta == $colEposta) {
        $guncelle = mysqli_prepare($con,"UPDATE kullanicilar SET isim=?, eposta=?
WHERE eposta=?");
        mysqli_stmt_bind_param($guncelle, "sss", $isim,$eposta,$eski_ePosta);
        mysqli_stmt_execute($guncelle);

        $response["success"] = true;
    }
}
echo json_encode($response);
?>

```

changePassword.php

```

<?php
require("password.php");

$con = mysqli_connect("localhost:3306", "cancaliskan", "Ee6!r3m7", "cancaliskan_toDoList");

$eposta = $_POST["eposta"];
$parola = $_POST["parola"];

$stmt = mysqli_prepare($con, "SELECT * FROM kullanicilar WHERE eposta = ?");
mysqli_stmt_bind_param($stmt, "s", $eposta);
mysqli_stmt_execute($stmt);
mysqli_stmt_store_result($stmt);
mysqli_stmt_bind_result($stmt, $colID, $colIsim, $colEposta, $colParola);

$response = array();
$response["success"] = false;

```

```

while(mysqli_stmt_fetch($statement)){
    $passwordHash = password_hash($parola, PASSWORD_DEFAULT);
    $guncelle = mysqli_prepare($con,"UPDATE kullanicilar SET parola=? WHERE eposta=?");
    mysqli_stmt_bind_param($guncelle, "ss", $passwordHash,$eposta);
    mysqli_stmt_execute($guncelle);
    $response["success"] = true;
}
echo json_encode($response);
?>

```

yapilacakEkle.php

```

<?php
    $con = mysqli_connect("localhost:3306", "cancaliskan", "Ee6!r3m7", "cancaliskan_toDoList");

    $eposta = $_POST["eposta"];
    $baslik = $_POST["baslik"];
    $metin = $_POST["metin"];
    $hatirlat = $_POST["hatirlat"];
    $tarih = $_POST["tarih"];
    $saat = $_POST["saat"];

    $statement = mysqli_prepare($con, "INSERT INTO yapilacaklar (eposta, baslik,
metin,hatirlat,tarih,saat) VALUES (?, ?, ?, ?, ?, ?)");
    mysqli_stmt_bind_param($statement, "ssssss", $eposta,$baslik,$metin,$hatirlat,$tarih,$saat);
    mysqli_stmt_execute($statement);
    mysqli_stmt_store_result($statement);
    //mysqli_stmt_bind_result($statement, $colID, $colIsim, $colEposta, $colParola);
    $response["success"] = true;

    echo json_encode($response);

```

?>

yapilacakGuncelle.php

<?php

```
$con = mysqli_connect("localhost:3306", "cancaliskan", "Ee6!r3m7", "cancaliskan_toDoList");
```

```
$ID = $_POST["ID"];
```

```
$baslik = $_POST["baslik"];
```

```
$metin = $_POST["metin"];
```

```
$hatirlat = $_POST["hatirlat"];
```

```
$tarih = $_POST["tarih"];
```

```
$saat = $_POST["saat"];
```

```
$response = array();
```

```
$response["success"] = false;
```

```
$guncelle = mysqli_prepare($con,"UPDATE yapılacaklar SET  
baslik=?,metin=?,hatirlat=?,tarih=?,saat=? WHERE ID=?");
```

```
mysqli_stmt_bind_param($guncelle, "ssssss",$baslik,$metin,$hatirlat,$tarih,$saat ,$ID);
```

```
mysqli_stmt_execute($guncelle);
```

```
$response["success"] = true;
```

```
/* $statement = mysqli_prepare($con, "SELECT * FROM yapılacaklar WHERE ID = ?");
```

```
mysqli_stmt_bind_param($statement, "s", $ID);
```

```
mysqli_stmt_execute($statement);
```

```
mysqli_stmt_store_result($statement);
```

```
mysqli_stmt_bind_result($statement, $colID, $colEposta, $colBaslik, $colMetin,$colHatirlat,  
$colTarih, $colSaat);
```



```

$response = array();

$response["success"] = false;

while(mysqli_stmt_fetch($statement)){*/

// }

echo json_encode($response);
?>

```

yapilacaklarListesi.php

```

<?php

$dsn = 'mysql:host=localhost;dbname=cancaliskan_toDoList;port=3306';
$usr = 'cancaliskan';
$pwd = 'Ee6!r3m7';
$db = new PDO($dsn, $usr, $pwd);
$db->setAttribute(PDO::ATTR_DEFAULT_FETCH_MODE, PDO::FETCH_ASSOC);

$eposta = $_POST["eposta"];

$query = $db->prepare('SELECT * FROM yapılacaklar WHERE eposta = :eposta');
$query->execute(array( 'eposta' => $eposta ));
$response["yapilacaklarListesi"] = $query->fetchAll();

echo json_encode($response);

```

References

- [1] <http://www.wiki-zero.net/index.php?q=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnL3dpa2kvSGFuZ21hbl8oZ2FtZSk>
- [2] <https://www.jetbrains.com/help/pycharm/configuring-python-interpreter.html>
- [3] <https://stackoverflow.com/questions/3786881/what-is-a-method-in-python>
- [4] https://www.tutorialspoint.com/python/python_functions.htm
- [5] <http://www.pythonforbeginners.com/code-snippets-source-code/game-hangman>
- [6] <https://gist.github.com/DevDarren/4199441>
- [7] <https://github.com/first20hours/google-10000-english>