```
import React, { useState, useEffect, useRef } from 'react';
import { initializeApp } from 'firebase/app';
import {
  getAuth,
  signInAnonymously,
  onAuthStateChanged,
  signOut,
  updateProfile
} from 'firebase/auth';
import {
  getFirestore,
  collection,
  addDoc,
  onSnapshot,
  doc,
  setDoc,
  getDoc,
  serverTimestamp
} from 'firebase/firestore';
import {
  Camera,
  Home,
  User,
  PlusSquare,
  Heart,
  MessageCircle,
  Send,
  LogOut,
  Image as ImageIcon,
  ArrowLeft,
  Loader2
} from 'lucide-react';

// --- Firebase Configuration ---
const firebaseConfig = JSON.parse(__firebase_config);
const app = initializeApp(firebaseConfig);
const auth = getAuth(app);
const db = getFirestore(app);
const appId = typeof __app_id !== 'undefined' ? __app_id : 'default-app-id';

// --- Helper: Image Compression ---
const compressImage = (file) => {
  return new Promise((resolve) => {
    const reader = new FileReader();
    reader.readAsDataURL(file);
    reader.onload = (event) => {
      const img = new Image();
      img.src = event.target.result;
      img.onload = () => {
        const canvas = document.createElement('canvas');
        const MAX_WIDTH = 600;
        const scaleSize = MAX_WIDTH / img.width;
        canvas.width = MAX_WIDTH;
        canvas.height = img.height * scaleSize;

        const ctx = canvas.getContext('2d');
        ctx.drawImage(img, 0, 0, canvas.width, canvas.height);

        resolve(canvas.toDataURL('image/jpeg', 0.6));
      };
    };
  });
};

// --- Components ---

function LoginScreen({ onLogin }) {
  const [mobile, setMobile] = useState('');
  const [otp, setOtp] = useState('');
  const [step, setStep] = useState(1);
  const [loading, setLoading] = useState(false);

  const handleSendOtp = () => {
    if (mobile.length < 10) return alert("Please enter valid mobile number");
    setLoading(true);
    // Simulate API delay
    setTimeout(() => {
      setStep(2);
      setLoading(false);
    }, 1000);
  };

  const handleVerifyOtp = () => {
    if (otp.length !== 4) return alert("Enter 4 digit OTP (Any 4 digits for demo)");
    setLoading(true);
    signInAnonymously(auth).catch((error) => {
      console.error(error);
      setLoading(false);
    });
  };

  return (
    <div className="flex flex-col items-center justify-center h-screen bg-white p-6">
      <h1 className="text-4xl font-cursive mb-8 text-pink-600" style={{fontFamily: 
'cursive'}}>Hey Chat</h1>

      <div className="w-full max-w-sm">
        {step === 1 ? (
          <>
            <label className="block text-gray-700 text-sm font-bold mb-2">Mobile
```

```
Number</label>
            <input
              type="tel"
              value={mobile}
              onChange={(e) => setMobile(e.target.value)}
              placeholder="+91 98765 43210"
              className="w-full border border-gray-300 rounded-lg px-4 py-3 mb-4
focus:outline-none focus:border-pink-500"
            />
            <button
              onClick={handleSendOtp}
              disabled={loading}
              className="w-full bg-blue-500 text-white font-bold py-3 rounded-lg
hover:bg-blue-600 transition flex justify-center items-center"
            >
              {loading ? <Loader2 className="animate-spin" /> : "Send OTP"}
            </button>
          </>
        ) : (
          <>
            <label className="block text-gray-700 text-sm font-bold mb-2">Enter OTP</
label>
            <input
              type="text"
              value={otp}
              onChange={(e) => setOtp(e.target.value)}
              placeholder="1234"
              maxLength={4}
              className="w-full border border-gray-300 rounded-lg px-4 py-3 mb-4
focus:outline-none focus:border-pink-500 text-center tracking-widest text-xl"
            />
            <button
              onClick={handleVerifyOtp}
              disabled={loading}
              className="w-full bg-blue-500 text-white font-bold py-3 rounded-lg
hover:bg-blue-600 transition flex justify-center items-center"
            >
              {loading ? <Loader2 className="animate-spin" /> : "Verify & Login"}
            </button>
            <button
              onClick={() => setStep(1)}
              className="w-full mt-4 text-sm text-gray-500 hover:text-gray-700"
            >
              Change Number
            </button>
          </>
        )}
      </div>
      <div className="mt-8 text-xs text-gray-400">
        By logging in, you agree to our Terms.
      </div>
    </div>
  );
}

function ProfileSetup({ user, onComplete }) {
  const [username, setUsername] = useState('');
  const [loading, setLoading] = useState(false);

  const handleSave = async () => {
    if (!username.trim()) return;
    setLoading(true);

    try {
      const userRef = doc(db, 'artifacts', appId, 'users', user.uid, 'profile', 'info');
      await setDoc(userRef, {
        username: username.toLowerCase().replace(/\s/g, '_'),
        uid: user.uid,
        joinedAt: serverTimestamp()
      });
      onComplete();
    } catch (error) {
      console.error("Error saving profile:", error);
      setLoading(false);
    }
  };

  return (
    <div className="flex flex-col items-center justify-center h-screen bg-white p-6">
      <h2 className="text-2xl font-bold mb-2">Create Username</h2>
      <p className="text-gray-500 mb-6 text-center">Pick a username for your new
account.</p>

      <div className="w-full max-w-sm">
        <div className="relative">
          <span className="absolute left-4 top-3 text-gray-400">@</span>
          <input
            type="text"
            value={username}
            onChange={(e) => setUsername(e.target.value)}
            placeholder="username"
            className="w-full border border-gray-300 rounded-lg pl-8 pr-4 py-3 mb-4
focus:outline-none focus:border-pink-500 lowercase"
          />
        </div>
        <button
          onClick={handleSave}
          disabled={loading || !username}
          className="w-full bg-blue-500 text-white font-bold py-3 rounded-lg
disabled:opacity-50"
```

```
        >
          {loading ? "Saving..." : "Next"}
        </button>
      </div>
    </div>
  );
}

function PostCard({ post }) {
  const [liked, setLiked] = useState(false);

  return (
    <div className="bg-white mb-4 border-b border-gray-100 pb-2">
      {/* Header */}
      <div className="flex items-center p-3">
        <div className="w-8 h-8 rounded-full bg-gradient-to-tr from-yellow-400 to-
pink-600 p-[2px]">
          <div className="w-full h-full rounded-full bg-white flex items-center justify-
center">
            <User size={16} className="text-gray-400" />
          </div>
        </div>
        <span className="ml-3 font-semibold text-sm">{post.username || 'user'}</
span>
      </div>

      {/* Image */}
      <div className="w-full aspect-square bg-gray-100 overflow-hidden">
        <img src={post.imageUrl} alt="Post" className="w-full h-full object-cover" />
      </div>

      {/* Actions */}
      <div className="flex items-center px-3 py-2 space-x-4">
        <button onClick={() => setLiked(!liked)}>
          <Heart size={24} className={liked ? "fill-red-500 text-red-500" : "text-
gray-700"} />
        </button>
        <MessageCircle size={24} className="text-gray-700" />
        <Send size={24} className="text-gray-700" />
      </div>

      {/* Caption */}
      <div className="px-3 pb-2">
        <p className="text-sm">
          <span className="font-semibold mr-2">{post.username}</span>
          {post.caption}
        </p>
        <p className="text-xs text-gray-400 mt-1 uppercase">
          {post.timestamp?.seconds ? new Date(post.timestamp.seconds *
1000).toLocaleDateString() : 'Just now'}
```

```
        </p>
      </div>
    </div>
  );
}

function UploadModal({ user, userProfile, onClose }) {
  const [image, setImage] = useState(null);
  const [caption, setCaption] = useState('');
  const [loading, setLoading] = useState(false);
  const fileInputRef = useRef(null);

  const handleFileChange = async (e) => {
    if (e.target.files[0]) {
      const compressed = await compressImage(e.target.files[0]);
      setImage(compressed);
    }
  };

  const handlePost = async () => {
    if (!image) return;
    setLoading(true);

    try {
      const postsRef = collection(db, 'artifacts', appId, 'public', 'data', 'posts');
      await addDoc(postsRef, {
        imageUrl: image,
        caption: caption,
        userId: user.uid,
        username: userProfile?.username || 'user',
        likes: 0,
        timestamp: serverTimestamp()
      });
      setLoading(false);
      onClose();
    } catch (error) {
      console.error("Error adding post: ", error);
      setLoading(false);
      alert("Failed to upload. Try a smaller image.");
    }
  };

  return (
    <div className="fixed inset-0 bg-white z-50 flex flex-col">
      <div className="flex items-center justify-between p-4 border-b">
        <button onClick={onClose}><ArrowLeft size={24} /></button>
        <h2 className="font-semibold text-lg">New Post</h2>
        <button
          onClick={handlePost}
          disabled={!image || loading}
          className="text-blue-500 font-semibold disabled:opacity-50"
        >
          {loading ? "Sharing..." : "Share"}
        </button>
      </div>

      <div className="flex-1 overflow-y-auto p-4">
        {!image ? (
          <div
            onClick={() => fileInputRef.current.click()}
            className="w-full aspect-square bg-gray-100 rounded-lg flex flex-col items-
center justify-center cursor-pointer border-2 border-dashed border-gray-300
hover:bg-gray-50"
          >
            <ImageIcon size={48} className="text-gray-400 mb-2" />
            <span className="text-gray-500 font-medium">Select Photo</span>
            <input
              type="file"
              accept="image/*"
              ref={fileInputRef}
              className="hidden"
              onChange={handleFileChange}
            />
          </div>
        ) : (
          <div className="flex flex-col space-y-4">
            <div className="w-full aspect-square rounded-lg overflow-hidden shadow-
sm">
              <img src={image} alt="Preview" className="w-full h-full object-cover" />
            </div>
            <button
              onClick={() => setImage(null)}
              className="text-red-500 text-sm font-medium self-start"
            >
              Remove photo
            </button>
            <textarea
              placeholder="Write a caption..."
              value={caption}
              onChange={(e) => setCaption(e.target.value)}
              className="w-full p-3 border rounded-lg focus:outline-none focus:border-
gray-400 resize-none h-24"
            />
          </div>
        )}
      </div>
    </div>
  );
}
```

```
// --- Main App Component ---

export default function HeyChatApp() {
  const [user, setUser] = useState(null);
  const [userProfile, setUserProfile] = useState(null);
  const [loading, setLoading] = useState(true);
  const [posts, setPosts] = useState([]);
  const [currentTab, setCurrentTab] = useState('home'); // home, profile
  const [showUpload, setShowUpload] = useState(false);

  // 1. Auth Listener
  useEffect(() => {
    const unsubscribe = onAuthStateChanged(auth, (currentUser) => {
      setUser(currentUser);
      if (!currentUser) setLoading(false);
    });
    return () => unsubscribe();
  }, []);

  // 2. Fetch User Profile
  useEffect(() => {
    if (!user) {
      setUserProfile(null);
      return;
    }

    // Check if profile exists
    const userRef = doc(db, 'artifacts', appId, 'users', user.uid, 'profile', 'info');
    const unsubProfile = onSnapshot(userRef, (docSnap) => {
      if (docSnap.exists()) {
        setUserProfile(docSnap.data());
      } else {
        setUserProfile(null); // Triggers profile setup screen
      }
      setLoading(false);
    });

    return () => unsubProfile();
  }, [user]);

  // 3. Fetch Feed Posts
  useEffect(() => {
    if (!user) return;

    const postsRef = collection(db, 'artifacts', appId, 'public', 'data', 'posts');
    const unsubPosts = onSnapshot(postsRef, (snapshot) => {
      const postsData = snapshot.docs.map(doc => ({
        id: doc.id,
        ...doc.data()
      }));
      // Sort client-side because of restriction on complex queries
      postsData.sort((a, b) => {
        const timeA = a.timestamp?.seconds || 0;
        const timeB = b.timestamp?.seconds || 0;
        return timeB - timeA;
      });
      setPosts(postsData);
    });

    return () => unsubPosts();
  }, [user]);

  if (loading) {
    return (
      <div className="flex items-center justify-center h-screen bg-white">
        <div className="text-center">
          <img src="https://upload.wikimedia.org/wikipedia/commons/a/a5/
Instagram_icon.png" width="80" className="mx-auto mb-4 animate-pulse
opacity-50" />
          <Loader2 className="animate-spin text-gray-400 mx-auto" />
        </div>
      </div>
    );
  }

  // Not Logged In
  if (!user) {
    return <LoginScreen />;
  }

  // Profile not set
  if (!userProfile) {
    return <ProfileSetup user={user} onComplete={() => {}} />;
  }

  // Upload Screen Overlay
  if (showUpload) {
    return <UploadModal user={user} userProfile={userProfile} onClose={() =>
setShowUpload(false)} />;
  }

  // Main App Interface
  return (
    <div className="max-w-md mx-auto bg-white h-screen flex flex-col shadow-2xl
relative">
      {/* Top Navbar */}
      <div className="px-4 py-3 flex items-center justify-between border-b bg-white
sticky top-0 z-10">
        <h1 className="text-2xl font-cursive font-bold select-none cursor-pointer"
style={{fontFamily: 'cursive'}}>Hey Chat</h1>
```

```
      <div className="flex space-x-4">
        <Heart size={24} className="text-gray-800" />
        <MessageCircle size={24} className="text-gray-800" />
      </div>
    </div>

    {/* Main Content Area */}
    <div className="flex-1 overflow-y-auto bg-white">
      {currentTab === 'home' && (
        <div className="pb-16">
          {/* Stories (Mock) */}
          <div className="flex space-x-4 p-4 overflow-x-auto border-b hide-
scrollbar">
            {['Your Story', 'rohit', 'priya', 'amit', 'sara'].map((name, i) => (
              <div key={i} className="flex flex-col items-center flex-shrink-0 space-
y-1">
                <div className={`w-16 h-16 rounded-full p-[2px] ${i === 0 ? 'bg-
gray-200' : 'bg-gradient-to-tr from-yellow-400 to-pink-600'}`}>
                  <div className="w-full h-full bg-white rounded-full flex items-center
justify-center overflow-hidden">
                    <User size={24} className="text-gray-300" />
                  </div>
                </div>
                <span className="text-xs truncate w-16 text-center">{name}</span>
              </div>
            ))}
          </div>

          {/* Posts Feed */}
          {posts.length === 0 ? (
            <div className="text-center py-20 text-gray-400">
              <Camera size={48} className="mx-auto mb-4 opacity-50" />
              <p>No posts yet.</p>
              <p className="text-sm">Be the first to upload!</p>
            </div>
          ) : (
            posts.map(post => <PostCard key={post.id} post={post} />)
          )}
        </div>
      )}

      {currentTab === 'profile' && (
        <div className="p-4">
          <div className="flex items-center justify-between mb-8">
             <h2 className="font-bold text-xl">{userProfile.username}</h2>
             <button onClick={() => signOut(auth)}><LogOut size={20}
className="text-red-500" /></button>
          </div>

          <div className="flex items-center mb-8">
```

```
            <div className="w-20 h-20 bg-gray-200 rounded-full flex items-center
justify-center mr-8">
              <User size={40} className="text-gray-400" />
            </div>
            <div className="flex space-x-8 text-center">
              <div>
                <div className="font-bold text-lg">{posts.filter(p => p.userId ===
user.uid).length}</div>
                <div className="text-sm text-gray-500">Posts</div>
              </div>
              <div>
                <div className="font-bold text-lg">124</div>
                <div className="text-sm text-gray-500">Followers</div>
              </div>
              <div>
                <div className="font-bold text-lg">365</div>
                <div className="text-sm text-gray-500">Following</div>
              </div>
            </div>
          </div>

          <div className="border-t pt-4 grid grid-cols-3 gap-1">
            {posts.filter(p => p.userId === user.uid).map(post => (
              <div key={post.id} className="aspect-square bg-gray-100 overflow-
hidden">
                <img src={post.imageUrl} className="w-full h-full object-cover" />
              </div>
            ))}
          </div>
        </div>
      )}
    </div>

    {/* Bottom Navigation */}
    <div className="h-14 bg-white border-t flex items-center justify-around fixed
bottom-0 w-full max-w-md pb-safe">
      <button onClick={() => setCurrentTab('home')} className={currentTab ===
'home' ? 'text-black' : 'text-gray-400'}>
        <Home size={28} strokeWidth={currentTab === 'home' ? 2.5 : 2} />
      </button>

      <button onClick={() => setShowUpload(true)} className="text-black">
        <PlusSquare size={28} />
      </button>

      <button onClick={() => setCurrentTab('profile')} className={currentTab ===
'profile' ? 'text-black' : 'text-gray-400'}>
        <div className={`w-7 h-7 rounded-full border border-gray-300 flex items-
center justify-center overflow-hidden ${currentTab === 'profile' ? 'ring-1 ring-black' : ''}
`}>
```

```
            <User size={16} />
          </div>
        </button>
      </div>
    </div>
  );
}
```