# Department of Computing

**CS-213: Advanced Programming**

**Name:** M. Hasnain Naeem

**Reg #:** 212728

**Class:** BSCS 7B

# Lab 12: React Native Login Application

**Date: 12 December, 2019**

**Time: 10:00-01:00pm & 02:00-05:00pm**

# Instructor: Dr. Sidra Sultana

**Lab Engineer: Ms. Ayesha Asif**

# 

# Lab 12: React Native Login Application

**Introduction**

React Native combines the best parts of native development with React, a best-in-class JavaScript library for building user interfaces.

**Objectives**

This lab will get students familiar with the React Native application Development.

**Tools/Software Requirement**

React native, Android Studio, JDK, node JS

**Description**

**Reference Videos**

<https://www.youtube.com/watch?v=0q3Jy0ZgzIw&list=PLYxzS__5yYQlHANFLwcsSzt3elIbYTG1h&index=23&t=0s>

<https://www.youtube.com/watch?v=6Z3bscC7p5Y&list=PLYxzS__5yYQlHANFLwcsSzt3elIbYTG1h&index=23>

<https://www.youtube.com/watch?v=StT6FZCBgN4&list=PLYxzS__5yYQlHANFLwcsSzt3elIbYTG1h&index=24>

<https://www.youtube.com/watch?v=2Uq3B3lskPM>

<https://www.youtube.com/watch?v=2Ok2dRg5vlk&list=PLYxzS__5yYQlHANFLwcsSzt3elIbYTG1h&index=26>

<https://www.youtube.com/watch?v=LfP7d09xjpI&list=PLYxzS__5yYQlHANFLwcsSzt3elIbYTG1h&index=27>

**Lab Task**

Create a login Application in react native

|  |
| --- |
| Solution |
| Task Code:  Home.js   |  | | --- | | *import* React, {Component} *from* 'react'; *import* {View, Text, TextInput, StyleSheet, Alert} *from* 'react-native';  *import* Heading *from* './Heading'; *import* Label *from* './Label'; *import Button from* './Button';   *export default class* Home *extends* Component {  state = {username: '', password: ''};   *static navigationOptions* = {header: *null*};   loginAction(status) {  *if*(status === 200)  *this*.props.navigation.navigate('Dashboard');  *else* Alert.*alert*('Error', 'Username or password did not match.', [{text: 'Okay'}]);  }   handleLogin() {  *const* {username, password} = *this*.state;  (*async* () => {  *let* response = *await fetch*('http://localhost:3000/login', {  method: 'POST',  headers: {  'Accept': 'application/json',  'Content-Type': 'application/json'  },  body: JSON.stringify({username: username, password: password})  });  *this*.loginAction(response.status);  })();  }   *handleUsernameChange* = (username)=> {  *this*.setState({username: username})  };   *handlePasswordChange* = (password)=> {  *this*.setState({password: password})  };   render() {  *return* (  <View *style*={styles.container}>  <Heading>Login</Heading>  <Label>Username</Label>  <View *style*={styles.inputContainer}>  <TextInput  *value*={*this*.state.username}  *style*={styles.input}  *placeholderTextColor*='#CACACA'  *selectionColor*='#666666'  *onChangeText*={*this*.*handleUsernameChange*}  />  </View>  <Label>Password</Label>  <View *style*={styles.inputContainer}>  <TextInput  *value*={*this*.state.password}  *style*={styles.input}  *placeholderTextColor*='#CACACA'  *selectionColor*='#666666'  *onChangeText*={*this*.*handlePasswordChange*}  *secureTextEntry*={*true*}  />  </View>  <Button *text*="Login" *handlePress*={() => *this*.handleLogin()}/>  </View>  );  } }  *const* styles = StyleSheet.create({  container: {  flex: 1,  justifyContent: 'center',  },  inputContainer: {  marginLeft: 40,  marginRight: 40,  shadowOpacity: 0.2,  shadowRadius: 3,  shadowColor: '#000000',  shadowOffset: { width: 2, height: 2 }  },  input: {  borderRadius: 20,  borderWidth: 1,  borderColor: '#2796ff',  paddingLeft: 20,  paddingRight: 20,  } }); |   Dashboard:   |  | | --- | | *import* React, {Component} *from* 'react'; *import* {View, Text, StyleSheet} *from* 'react-native';  *export default class* Home *extends* Component {  render() {  *return* (  <View *style*={styles.container}>  <Text>Welcome to admin panel.</Text>  </View>  );  } }   *const* styles = StyleSheet.create({  container: {  marginLeft: 40,  marginTop: 40  }, }); |   Button:   |  | | --- | | *import* React *from* 'react' *import* { View, Text, StyleSheet, TouchableHighlight } *from* 'react-native'  *const Button* = ({ text, handlePress, isDisabled}) => {  *if*(isDisabled)  *return* (  <View *style*={styles.buttonContainer}>  <TouchableHighlight *style*={styles.disabledButton} *onPress*={handlePress}>  <Text *style*={styles.submit}>  {text}  </Text>  </TouchableHighlight>  </View>  );  *else return* (  <View *style*={styles.buttonContainer}>  <TouchableHighlight *underlayColor*='#2c63b6' *style*={styles.button} *onPress*={handlePress}>  <Text *style*={styles.submit}>  {text}  </Text>  </TouchableHighlight>  </View>  ); };  *const* styles = StyleSheet.create({  buttonContainer: {  alignItems: 'flex-end'  },  button: {  height: 50,  paddingLeft: 20,  paddingRight: 20,  backgroundColor: '#2796ff',  width: 200,  marginRight: 120,  marginTop: 15,  borderWidth: 1,  borderColor: 'rgba(0,0,0,.1)',  justifyContent: 'center',  alignItems: 'center',  borderRadius: 500  },  disabledButton: {  height: 50,  paddingLeft: 20,  paddingRight: 20,  backgroundColor: 'black',  width: 200,  marginRight: 120,  marginTop: 15,  borderWidth: 1,  borderColor: 'rgba(0,0,0,.1)',  justifyContent: 'center',  alignItems: 'center',  borderRadius: 500  },  submit: {  color: 'white',  fontWeight: '600'  } })  *export default Button* |   Label.js   |  | | --- | | *import* React *from* 'react' *import* { Text, StyleSheet } *from* 'react-native'  *class* Label *extends* React.Component {  render(){  *return* <Text *style*={styles.label}>{*this*.props.children}</Text>;  } }  *const* styles = StyleSheet.create({  label: {  marginTop: 5,  marginBottom: 5,  fontWeight: '100',  color: "#2796ff",  textAlign: 'center',  } });   *export default* Label; |   Heading.js   |  | | --- | | *import* React *from* 'react' *import* { View, Text, StyleSheet } *from* 'react-native'  *class* Heading *extends* React.Component {  render() {  *return* (  <View *style*={styles.header}>  <Text *style*={styles.headerText}>  {*this*.props.children}  </Text>  </View>  )  } } *const* styles = StyleSheet.create({  header: {  marginBottom: 20  },  headerText: {  textAlign: 'center',  fontSize: 50,  color: "#2796ff",  fontWeight: '400',  } });  *export default* Heading |   Server Side  App.js   |  | | --- | | *var createError* = *require*('http-errors'); *var express* = *require*('express'); *var* path = require('path'); *var cookieParser* = require('cookie-parser'); *var logger* = require('morgan');  *var* indexRouter = *require*('./routes/index'); *var* loginRouter = require('./routes/login');  *var* app = *express*();  *// view engine setup* app.set('views', path.*join*(\_\_dirname, 'views')); app.set('view engine', 'pug');  app.use(*logger*('dev')); app.use(*express*.json()); app.use(*express*.urlencoded({ extended: *false* })); app.use(*cookieParser*()); app.use(*express*.static(path.*join*(\_\_dirname, 'public')));  app.use('/', indexRouter); app.use('/login', loginRouter);  *// catch 404 and forward to error handler* app.use(*function*(req, res, next) {  next(*createError*(404)); });  *// error handler* app.use(*function*(err, req, res, next) {  *// set locals, only providing error in development* res.locals.message = err.message;  res.locals.error = req.app.get('env') === 'development' ? err : {};   *// render the error page* res.status(err.status || 500);  res.render('error'); });  module.exports = app; |   Login.js   |  | | --- | | *var express* = require('express'); *var* router = *express*.Router();  */\* GET users listing. \*/* router.post('/', *function*(req, res, next) {  console.log(req.body);  *try* {  *let* details = req.body;  *if* (details.username === "Admin" && details.password === "Admin")  res.status(200).json({response: "pass"});  *else* res.status(300).json({response: "pass"});  }  *catch*(e){  res.status(300).json({response: "fail"});  } });  module.exports = router; |   Task Output Screenshot:  https://scontent-kut2-2.xx.fbcdn.net/v/t1.15752-9/78910690_990877541269348_4994508127240978432_n.jpg?_nc_cat=102&_nc_ohc=twxFZII5JuAAQlsf1-C1buDvcJwRPq0r6yqYg6a3e1RhOMS3v39rj8EHg&_nc_ht=scontent-kut2-2.xx&oh=0aab35e59bca619485aa5fac1b94918f&oe=5E83B45E  https://scontent-kut2-2.xx.fbcdn.net/v/t1.15752-9/79645803_623130101775208_5290644966953975808_n.jpg?_nc_cat=102&_nc_ohc=GZbgmfmUCt0AQnwdSjcZB1f82BRfGlpr4NCWkXhGLWrweND0nne6r7abg&_nc_ht=scontent-kut2-2.xx&oh=6a96cab72675ea5235f7978b9e96fedf&oe=5E7B18E1  https://scontent-kut2-2.xx.fbcdn.net/v/t1.15752-9/79321570_564089931041011_3648364152885346304_n.jpg?_nc_cat=110&_nc_ohc=XjYEOZF44vQAQmxzDlzuOMslgIi4JFkv0ODe3CcIuLBLODfeV0hnmJSQg&_nc_ht=scontent-kut2-2.xx&oh=2b7df1252b4122df8249e1a6c2e0343b&oe=5E696EDD  Server Side: |

### Deliverable

Compile a single word document by filling in the solution part and submit this Word file on LMS. This lab grading policy is as follows: The lab is graded between 0 to 10 marks. The submitted solution can get a maximum of 5 marks. At the end of each lab or in the next lab, there will be a viva/quiz related to the tasks. You must show the implementation of the tasks in the designing tool, along with your complete Word document to get your work graded. You must also submit this Word document on the LMS. In case of any problems with submissions on LMS, submit your Lab assignments by emailing it to Ms. Ayesha Asif: [ayesha.asif@seecs.edu.pk](mailto:ayesha.asif@seecs.edu.pk).