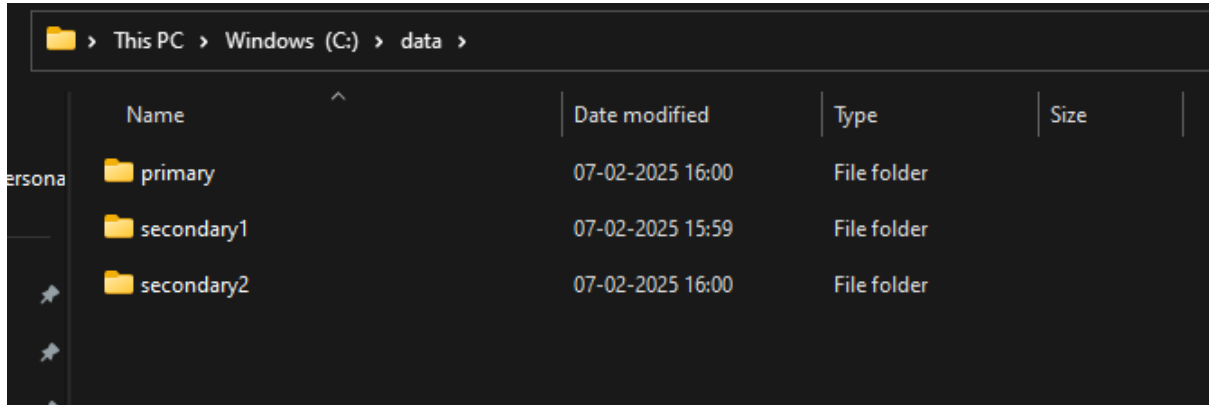


PRACTICAL NO. 6 PRADNYA TAGAD L037

Replication using MongoDB

Step 1: Create folder in c drive



Step 2: Go to Windows PowerShell ISE>New file(make three of them/ type same command with different port and path)

Commands:

```
mongod --port=2717 --dbpath="C:\data\primary" --replSet="test-replica-set"
mongod --port=2727 --dbpath = "C:\data\secondary1" --replSet="test-replica-set"
mongod --port=2737 --dbpath = "C:\data\secondary2" --replSet="test-replica-set"
```

The screenshot shows a Windows PowerShell terminal window. The user has entered the command: `PS C:\Users\Admin> mongod --port=2717 --dbpath = "C:\data\primary" --replSet="test-replica-set"`. The output shows the MongoDB server starting up, displaying various logs and configuration details. The terminal text is as follows:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Admin> mongod --port=2717 --dbpath = "C:\data\primary" --replSet="test-replica-set"
{"t":{"sdate":"2025-02-07T15:10:51.173+05:30"},"s":"I",  "c":"NETWORK",  "id":4915701, "ctx":"","msg":"Initialized wire specification","attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":17},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":17},"outgoing":{"minWireVersion":6,"maxWireVersion":17},"isInternalClient":true}}}}
{"t":{"sdate":"2025-02-07T15:10:52.613+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,  "ctx":"","msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"sdate":"2025-02-07T15:10:52.614+05:30"},"s":"I",  "c":"NETWORK",  "id":4648602, "ctx":"thread1","msg":"Implicit TCP FastOpen in use."}
{"t":{"sdate":"2025-02-07T15:10:52.614+05:30"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"thread1","msg":"Successfully registered PrimaryOnlyService","attr":{"service":"TenantMigrationDonorService","namespace":"config.tenantMigrationDonors"}}
{"t":{"sdate":"2025-02-07T15:10:52.616+05:30"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"thread1","msg":"Successfully registered PrimaryOnlyService","attr":{"service":"TenantMigrationRecipientService","namespace":"config.tenantMigrationRecipients"}}
{"t":{"sdate":"2025-02-07T15:10:52.616+05:30"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"thread1","msg":"Successfully registered PrimaryOnlyService","attr":{"service":"ShardSplitDonorService","namespace":"config.tenantSplitDonors"}}
Invalid command: C:\data\primary
Options:
  --networkMessageCompressors arg (snappy,zstd,zlib)
                                Comma-separated list of compressors to
                                use for network messages

General options:
  -h [ -help ]                  Show this usage information
  --version                     Show version information
  -f [ --config ] arg           Configuration file specifying
                                additional options
  --configExpand arg            Process expansion directives in config
                                file (none, exec, rest)
  --port arg                    Specify port number - 27017 by default
  --ipv6                        Enable IPv6 support (disabled by
                                default)
  --listenBacklog arg (-2147483647) Set socket listen backlog size
  --maxConns arg (-1000000)      Max number of simultaneous connections
  --pidfilepath arg             Full path to pidfile (if not set, no
                                pidfile is created)
  --timeZoneInfo arg            Full path to time zone info directory,
                                e.g. /usr/share/zoneinfo
  -v [ --verbose ] [=arg(s)]    Be more verbose (include multiple times
                                for more verbosity e.g. -vvvvv)
  --quiet                       Quieter output
  --logpath arg                 Log file to send write to instead of
                                stdout - has to be a file, not
                                directory
  --logappend                   Append to logpath instead of
                                over-writing
  --logRotate arg               Set the log rotation behavior
                                (rename|reopen)
  --timestampFormat arg         Desired format for timestamps in log
                                messages. One of iso8601-utc or
                                iso8601-local
```

```
PS C:\Users\Admin> mongod --port=2717 --dbpath="C:\data\primary" --replSet="test-replica-set"
{"t":{"$date":"2025-02-07T15:14:42.021+05:30"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"","msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}}
{"t":{"$date":"2025-02-07T15:14:42.022+05:30"},"s":"I", "c":"NETWORK", "id":4915791, "ctx":"","msg":"Initialized wire specification", "attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":17},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":17},"outgoing":{"minWireVersion":6,"maxWireVersion":17},"isInternalClient":true}}}}
{"t":{"$date":"2025-02-07T15:14:43.400+05:30"},"s":"I", "c":"NETWORK", "id":4608602, "ctx":"thread1", "msg":"Implicit TCP FastOpen in use."}}
{"t":{"$date":"2025-02-07T15:14:43.412+05:30"},"s":"I", "c":"REPL", "id":5123808, "ctx":"thread1", "msg":"Successfully registered PrimaryOnlyService", "attr":{"service":"TenantMigrationDonorService","namespace":"config.tenantMigrationDonors"}}
{"t":{"$date":"2025-02-07T15:14:43.412+05:30"},"s":"I", "c":"REPL", "id":5123808, "ctx":"thread1", "msg":"Successfully registered PrimaryOnlyService", "attr":{"service":"TenantMigrationRecipientService","namespace":"config.tenantMigrationRecipients"}}
{"t":{"$date":"2025-02-07T15:14:43.412+05:30"},"s":"I", "c":"REPL", "id":5123808, "ctx":"thread1", "msg":"Successfully registered PrimaryOnlyService", "attr":{"service":"ShardSplitDonorService","namespace":"config.tenantSplitDonors"}}
{"t":{"$date":"2025-02-07T15:14:43.412+05:30"},"s":"I", "c":"CONTROL", "id":5945603, "ctx":"thread1", "msg":"Multi threading initialized"}}
{"t":{"$date":"2025-02-07T15:14:43.415+05:30"},"s":"I", "c":"CONTROL", "id":4615611, "ctx":"","msg":"MongoDB starting", "attr":{"pid":6776,"port":2717,"dbPath":"C:/data/primary","architecture":"64-bit","host":"MUM1515CPU1083"}}
{"t":{"$date":"2025-02-07T15:14:43.415+05:30"},"s":"I", "c":"CONTROL", "id":23398, "ctx":"","msg":"Target operating system minimum version", "attr":{"targetMinOS":"Windows 7/Windows Server 2008 R2"}}
{"t":{"$date":"2025-02-07T15:14:43.415+05:30"},"s":"I", "c":"CONTROL", "id":23403, "ctx":"","msg":"Build Info", "attr":{"buildInfo":{"version":"6.0.13","gitVersion":"3b13907f9bdf6bd3264d67148d6c215d51bbd28c","modules":[],"allocator":"tcmalloc","environment":{"distmod":"windows","distarch":"x86_64","target_arch":"x86_64"}}}}
{"t":{"$date":"2025-02-07T15:14:43.415+05:30"},"s":"I", "c":"CONTROL", "id":51765, "ctx":"","msg":"Operating System", "attr":{"os":{"name":"Microsoft Windows 10","version":"10.0 (build 22621)"}}}
{"t":{"$date":"2025-02-07T15:14:43.415+05:30"},"s":"I", "c":"CONTROL", "id":21951, "ctx":"","msg":"Options set by command line", "attr":{"options":{"net":{"port":2717,"replication":{"replSet":"test-replica-set"},"storage":{"dbPath":"C:/data/primary"}}}}}
{"t":{"$date":"2025-02-07T15:14:43.418+05:30"},"s":"I", "c":"STORAGE", "id":22315, "ctx":"","msg":"Opening WiredTiger", "attr":{"config":{"create,cache_size=3471M,session_max=33000,evict=(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,remove=true,path=journal,compressor=snappy),builtin_extension_config=(zstd=(compression_level=6)),file_manager=(close_idle_time=600,close_scan_interval=10,close_handle_minimum=2000),statistics_log=(wait=0),json_output=(error,message),verbose=(recovery_progress:1,checkpoint_progress:1,compact_progress:1,backup:0,checkpoint:0,compact:0,evict:0,history_store:0,recovery:0,rts:0,salvage:0,tiered:0,timestamp:0,transaction:0,verify:0,log:0)}}}}
{"t":{"$date":"2025-02-07T15:14:43.473+05:30"},"s":"I", "c":"STORAGE", "id":4795996, "ctx":"","msg":"WiredTiger opened", "attr":{"durationMillis":55}}
{"t":{"$date":"2025-02-07T15:14:43.473+05:30"},"s":"I", "c":"RECOVERY", "id":23397, "ctx":"","msg":"WiredTiger recoveryTimestamp", "attr":{"recoveryTimestamp":{"$timestamp":{"t":0,"i":0}}}}
{"t":{"$date":"2025-02-07T15:14:43.508+05:30"},"s":"W", "c":"CONTROL", "id":22120, "ctx":"","msg":"Access control is not enabled for the database. Read and write access to data and configuration is unrestricted", "tags":["startupWarnings"]}
{"t":{"$date":"2025-02-07T15:14:43.508+05:30"},"s":"W", "c":"CONTROL", "id":22140, "ctx":"","msg":"This server is bound to localhost. Remote systems will be unable to connect to this server. S
```

Step 3: New window

Commands:

```
mongosh --host="localhost:2717" (make new 2 windows with other localhost)
```

```
rs.initiate()
```

```
rs.add({"host:localhost:2727"});
```

```
rs.status()
```

```
PS C:\Users\Admin> mongosh --host="localhost:2717"
Current Mongosh Log ID: 67a5d860d3f1d8ba3a772d8c
Connecting to: mongod://localhost:2717/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.1.4
Using MongoDB: 6.0.13
Using Mongosh: 2.1.4
mongosh 2.3.8 is available for download: https://www.mongodb.com/try/download/shell

For mongosh info see: https://docs.mongodb.com/mongosh-shell/

-----
The server generated these startup warnings when booting
2025-02-07T15:14:43.508+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
2025-02-07T15:14:43.508+05:30: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip_all

-----

test> rs.initiate()
{
  info2: 'no configuration specified. Using a default configuration for the set',
  me: 'localhost:2717',
  ok: 1
}

test-replica-set [direct: primary] test> rs.add({"host:localhost:2727"});
Uncaught:
SyntaxError: Unexpected token, expected ",", (1:22)
```

Error: rs.add({"host:localhost:2727"});

```
test-replica-set [direct: primary] test> rs.status()
{
  set: 'test-replica-set',
  date: ISODate('2025-02-07T10:02:32.471Z'),
  myState: 1,
  term: Long('1'),
  syncSourceHost: '',
  syncSourceId: -1,
  heartbeatIntervalMillis: Long('2000'),
  majorityVoteCount: 2,
  writeMajorityCount: 2,
  votingMembersCount: 3,
  writableVotingMembersCount: 3,
  optimes: {
    lastCommittedOpTime: { ts: Timestamp({ t: 1738922548, i: 1 }), t: Long('1') },
    lastCommittedWallTime: ISODate('2025-02-07T10:02:28.864Z'),
    readConcernMajorityOpTime: { ts: Timestamp({ t: 1738922548, i: 1 }), t: Long('1') },
    appliedOpTime: { ts: Timestamp({ t: 1738922548, i: 1 }), t: Long('1') },
    durableOpTime: { ts: Timestamp({ t: 1738922548, i: 1 }), t: Long('1') },
    lastAppliedWallTime: ISODate('2025-02-07T10:02:28.864Z'),
    lastDurableWallTime: ISODate('2025-02-07T10:02:28.864Z')
  },
  ...
}
```

- I. show dbs
- II. use practice
- III. db.users.insert({name:"Shruti"})
- IV. db.users.find()

```
test-replica-set [direct: primary] test> show dbs
admin      80.00 KiB
config    160.00 KiB
local     404.00 KiB
test-replica-set [direct: primary] test> use practice
switched to db practice
test-replica-set [direct: primary] practice> db.users.insert({name:"Shruti"})
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('67a5dc0ad3f1d8ba3a772d8d') }
}
test-replica-set [direct: primary] practice> db.users.find()
[ { _id: ObjectId('67a5dc0ad3f1d8ba3a772d8d'), name: 'Shruti' } ]
test-replica-set [direct: primary] practice> |
```

Now repeat I, ii, iv for rest 2 windows

Localhost:2727

```
test-replica-set [direct: secondary] test> show dbs
admin      80.00 KiB
config    204.00 KiB
local     404.00 KiB
test-replica-set [direct: secondary] test> use practice
switched to db practice
```

```
test-replica-set [direct: secondary] practice> db.users.find()
[ { _id: ObjectId('67a5dc0ad3f1d8ba3a772d8d'), name: 'Shruti' } ]
test-replica-set [direct: secondary] practice> |
```

Localhost:2737

```
test-replica-set [direct: secondary] test> show dbs
admin      80.00 KiB
config    200.00 KiB
local     404.00 KiB
test-replica-set [direct: secondary] test> use practice
switched to db practice
```

```
test-replica-set [direct: secondary] practice> db.users.find()
MongoServerError[NotPrimaryNoSecondaryOk]: not primary - consider using db.getMongo().setReadPref() or readPreference in the connection string
test-replica-set [direct: secondary] practice> db.getMongo().setReadPref("secondaryPreferred")

test-replica-set [direct: secondary] practice> db.users.find()
[ { _id: ObjectId('67a5dc0ad3f1d8ba3a772d8d'), name: 'Shruti' } ]
test-replica-set [direct: secondary] practice> |
```